

Firmware pro sinusový měnič

Sine Wave Inverter Firmware

Zadání diplomové práce

Student:

Bc. Radim Kunčický

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Firmware pro sinusový měnič
Sine Wave Inverter Firmware

Zásady pro vypracování:

Cílem práce je navrhnout firmware sinusového měniče a následně prakticky ověřit jeho funkčnost. Jádrem řešení bude monolitický počítač připojený k H-můstku budící transformátor měniče. Firmware bude obsahovat regulaci PWM na základě zpětné vazby, komunikaci s počítačem pomocí sériové linky nebo USB s možností změny konfigurace výstupu za běhu.

1. Úvod k řízení PWM, H-můstky, řešerše obdobných zapojení.
2. Návrh a hardwarová realizace jednoduchého měniče.
3. Implementace firmwaru, otestování základní funkčnosti.
4. Návrh komunikace s počítačem, konfigurace a měření sledovaných parametrů.
5. Experimentální ověření funkčnosti a stability.
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] Barnett, R., Cox, S., O'Cull, L., Embedded C Programming and the Atmel AVR, Delmar Cengage Learning, 2006, ISBN 1418039594
- [2] Pressman, A., Billings, K., Morey, T., Switching Power Supply Design, McGraw-Hill Professional, 3rd Edition, 2009, ISBN 0071482725
- [3] Sueker, L. H., Power Electronics Design: A Practitioner's Guide, Newnes, 2005, ISBN 0750679271

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Ing. Michal Krumník**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

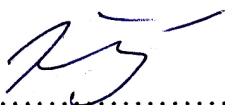


doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2014

.....

Rád bych na tomto místě poděkoval Ing. Michalu Krumníkovi za vedení mé diplomové práce a cenné rady, bez kterých by práce nevznikla.

Abstrakt

Cílem práce je zmapovat konstrukce pulzních měničů vhodných pro napájení zařízení závislých na rozvodné síti při výpadku. Popsat jejich topologie, způsob aproximace průběhu síťového napětí a zvláště pak možnosti regulace a řízení za pomoci moderních mikrokontrolérů. Jsou zde definovány jak elektrotechnické požadavky na takové zařízení tak požadavky na řízení a kvalitu řízení. V práci je popsán návrh jednoduchého měniče, vhodného pro experimentální vývoj firmware a následně detailně popsány možnosti a způsoby implementace řízení, snímání a komunikace pomocí 8 bitového mikrokontroléru firmy Atmel.

Klíčová slova: Pulzní měniče, regulace, UPS, Atmel, RS-232, firmware

Abstract

The aim is to describe the structure of pulse converters suitable for backing up devices sensitive to a power grid failure. The thesis describes their topology, sine wave approximation and especially the possibility of regulation and control using modern microcontrollers. There are various standards that define electrical requirements for such devices as the control and quality of control. The thesis discusses the design of a simple inverter suitable for experimental development of firmware and then describes in more details the possibilities and ways how to implement management, sensing and communication using an 8-bit microcontroller from Atmel.

Keywords: Pulse inverter, regulation, UPS, Atmel, RS-232, firmware

Seznam použitých zkratek a symbolů

ADC	– Analog to digital converter
AD, DA	– Analogově digitální, digitálně analogový
ALU	– Aritmeticko-logická jednotka
ASA	– American Standards Association
ASCII	– American Standard Code for Information Interchange
AVR	– Alf Vegard Risc procesor
DCE	– Data Communications Equipment
DSP	– Digitální signálový procesor
DTE	– Data Terminal Equipment
EEPROM	– Electrically Erasable Programmable Read-Only Memory
EMI	– Electromagnetic Interference
IO	– Integrovaný obvod
ISO	– International Organization for Standardization
LCD	– Liquid Crystal Display
log	– Logický / á
LUT	– Look Up Table
MCU	– Microcontroller Unit
MIPS	– Million Instruction Per Second
MOSFET	– Metal Oxide Semiconductor Field Effect Transistor
PC	– Personal Computer
PWM	– Pulse Width Modulation
PŠM	– Pulsní šířková modulace
RAM	– Random-access Memory
RISC	– Reduced Instruction Set Computing
TTL	– Tranzistorově-tranzistorová logika
UPS	– Uninterruptible Power Supply
UTF-8	– UCS Transformation Format

Obsah

1 Úvod	6
2 Pulzní měniče	7
2.1 Výhody a nevýhody	7
2.2 Části pulzních měničů	7
2.3 Topologie měničů	10
2.4 Rešerše obdobných zapojení	14
3 Pulzně šířková modulace	17
3.1 Principy PWM	17
3.2 Způsob modulace a demodulace	18
4 Regulace	19
4.1 Druhy regulace	19
4.2 Regulační členy	19
4.3 Regulátory	21
4.4 Číslicová regulace	22
4.5 Nastavení regulátoru	23
5 Návrh a realizace jednoduchého měniče	25
6 Implementace firmware	28
6.1 Atmel Mega644PA	28
6.2 Části firmware	30
6.3 Inicializace výstupu	30
6.4 Modulace sinu pomocí vysokofrekvenční PWM	31
6.5 Modulace 50 Hz PWM s mezikrokem	33
6.6 Generování polovin vln	34
6.7 Přejít mezi modulacemi	34
6.8 Měření proudu a napětí	35
6.9 Regulace výstupu	37
7 Návrh komunikace s počítačem, konfigurace a měření sledovaných parametrů	39
7.1 RS-232	39
7.2 Návrh komunikačního protokolu	41
7.3 Sériový port Atmel Mega 644PA a jeho obsluha	43
7.4 Display	44
8 Experimentální ověření funkčnosti a stability	46
9 Závěr	49
10 Reference	50

Přílohy	50
A Schéma elektrické části	51
B Testování komunikace	53
C Grafy a měření	54
D Architektura firmware	61
D.1 Vysoká priorita	61
D.2 Střední priorita	61
D.3 Nízká priorita	61
D.4 Funkce	61

Seznam tabulek

1	Nastavení PID regulátoru podle Ziegler–Nicholsovy metody	24
2	Počet strojových cyklů potřebných pro provedení operace a výsledná velikost programu [2].	30
3	Měřené veličiny a jejich rozsah.	35
4	Závislost délky vedení na modulační rychlosti [6].	40
5	Porovnání synchronního a asynchronního přenosu u RS-232	40
6	Zprávy, které je měnič schopen zpracovat	43

Seznam obrázků

1	Schéma vzorového měniče [3].	8
2	Topologie paralelní dvojčinný (push-pull)	12
3	Polomůstek (half-bridge)	12
4	Polomůstek (full-bridge)	13
5	Budič typu totem	14
6	Blokové schéma APC SmartUPS	15
7	Topologie UPS Microchip	15
8	Příklad závislosti střídý na hodnotě vstupního napětí	17
9	Blokové schéma modulátoru	18
10	Funkční schéma regulace	20
11	Odezva PT členu	20
12	Blokové schéma měniče	27
13	Blokové schéma mikrokontroléru Mega 644	29
14	Způsob generování PWM pomocí čítače a využívání registrů	31
15	Obsluha přijetí zprávy sériovým portem	45
16	Zákřivky způsobené transformátorem	47
17	Graf vybíjení baterie	48
A.18	Závislost snímané hodnoty proudu a hodnoty převodu ADC	54
A.19	Závislost snímané hodnoty proudu a hodnoty převodu ADC, aproximace přímkami	54
A.20	Výstup mikrokontroléru při vysokofrekvenční PWM. Průběh je vytvořen inverzí jednoho výstupu a následným součtem.	55
A.21	Napětí G elektrody při 50 Hz PWM	55
A.22	Napětí G elektrody při modulovaném sinu	56
A.23	Nevyfiltrovaný výstupní sinusový průběh	56
A.24	Vyfiltrovaný výstupní sinusový průběh	57
A.25	Výstupní průběh 50Hz PWM	57
A.26	Výstupní průběh 50Hz PWM, saturace jádra	58
A.27	Napětí D-S při 50Hz PWM	58
A.28	Napětí D-S při 50Hz PWM, saturace jádra	59
A.29	Napětí D-S při vysokofrekvenční PWM	59
A.30	Výstup budiče při vysokofrekvenční PWM	60

Seznam výpisů zdrojového kódu

1	Výpočet doby plnění	32
2	Naplnění pole s aproximací funkce sinus	32
3	Výpočet změny nastavovací hodnoty	37

1 Úvod

V posledních letech došlo k obrovskému rozvoji solárních elektráren, vlivem čehož cena fotovoltaických článků poklesla na úroveň, kdy jsou dostupné i běžným domácnostem. Se solárními panely pro výrobu elektrické energie se tak můžeme často setkat i na střechách rodinných domů.

Solární články mohou sloužit i v místech, kde není dostupná veřejná rozvodná síť. V době slunečního svitu jsou nabíjeny baterie a jejich energie je pak využívána v době potřeby. Napětí baterií ale není vhodné pro provozování naprosté většiny domácích spotřebičů a je nutno do řetězce včlenit další zařízení. Tyto zařízení - měniče jsou svým principem velmi podobné, ne-li stejné s nepřerušitelnými zdroji energie (UPS), které jsou používány pro osobní počítače.

Nutnost zvýšit napětí baterie vyžaduje použití měniče s výstupním napětím 230 V a frekvencí 50 Hz. Většina běžně dostupných UPS ale není schopna dosáhnout na výstupu sinusového průběhu, používají 50 Hz modifikovaný obdélník, který se snaží přiblížit sinusovému průběhu. Jednou z nevýhod takového průběhu je, že není vhodný pro provoz strojů, které jsou závislé na točivém magnetickém poli - například asynchronního motoru. V rodinném domě může být provoz asynchronního motoru regulérním požadavkem, příkladem může být oběhové čerpadlo topení v zimě, bez kterého může dojít ke škodám na topném systému.

Možností, jak dosáhnout vhodného tvaru, je použití vysokofrekvenční pulzní šířkové modulace, kterou budeme modulovat sinusový průběh. Taková konfigurace není zcela běžná a odborná literatura se jí věnuje jen okrajově nebo vůbec. Nejlepším zdrojem informací jsou referenční návrhy takových měničů pocházející od výrobců elektronických součástek jako je Texas Instruments nebo Microchip. Tyto návrhy jsou založeny většinou na analogových součástkách a digitální obvody - mikroprocesory hrají jen roli obvodů zajišťujících například komunikaci s okolím. Pokud je návrh postaven na mikroprocesoru nebo mikrokontroléru, jedná se většinou o výkoný čip, který je svoji architekturou velmi blízký DSP. Výrobci UPS se také drží velmi konzervativního designu, většinou postaveného okolo zakázkového analogového monolitického obvodu.

Vyvstává otázka, zda-li je taková konstrukce řešitelná pomocí dnešních moderních osmi nebo šestnácti bitových mikrokontrolérů a jsou-li schopny zvládnout řízení tak kritické aplikace, protože na chodu měniče mohou být závislé spotřebiče, které mohou svoji cenou mnohonásobně překračovat cenu samotného měniče. Výhodou takových MCU je hlavně jejich nízká cena srovnatelná s cenou konstrukce z analogových součástek, snadná dostupnost a možnost jednoduše replikovat konstrukce, které je využívají. Úprava funkce a rozšíření při změně požadavků je jen otázkou softwaru a nevyžaduje zásah do hardwaru.

Cílem práce je zhodnotit dostupné konstrukce podobných měničů, ověřit možnosti takové konstrukce, navrhnout jednoduchý měnič, který bude sloužit k vývoji firmwaru a následně navrhnout a realizovat software, na kterém bude možné demonstrovat jak možnosti měniče, tak možnosti mikrokontroléru v této aplikaci.

2 Pulzní měniče

Výběr zdroje napětí je velmi ovlivněn požadavky dané aplikace. Spínané a lineární zdroje mají různé výhody i nevýhody pro každý případ užití.

Lineární zdroje mají tři základní výhody. Velkou výhodou je jednoduchá konstrukce, stačí použít monolitický stabilizátor a několik součástek okolo. Další výhodou je tichý chod, jak po stránce akustické tak po stránce vyzařování elektromagnetického rušení. Konečnou výhodou, která často rozhodne pro použití lineárního zdroje je pro nízké příkony, v řádech jednotek wattů, nízká cena.

2.1 Výhody a nevýhody

Nevýhodou lineárních zdrojů je, že nemohou pracovat ve zvyšující konfiguraci. Vstupní napětí musí být vždy minimálně o 2-3 volty vyšší než výstupní. Z toho plyne i relativně malá účinnost takového zdroje, která se obvykle pohybuje mezi 30-60 %. To znamená, že pro každý watt dodaný do zátěže musíme do zdroje dodat wattů dva. Následkem nízké účinnosti je nutnost kvalitního chlazení, které musí být dimenzováno na maximální možné provozní zatížení. S velkou pravděpodobností, ale nebude v takových podmínkách zdroj běžně provozován, po většinu života budou tedy chladiče nevyužity a předimenzovány.

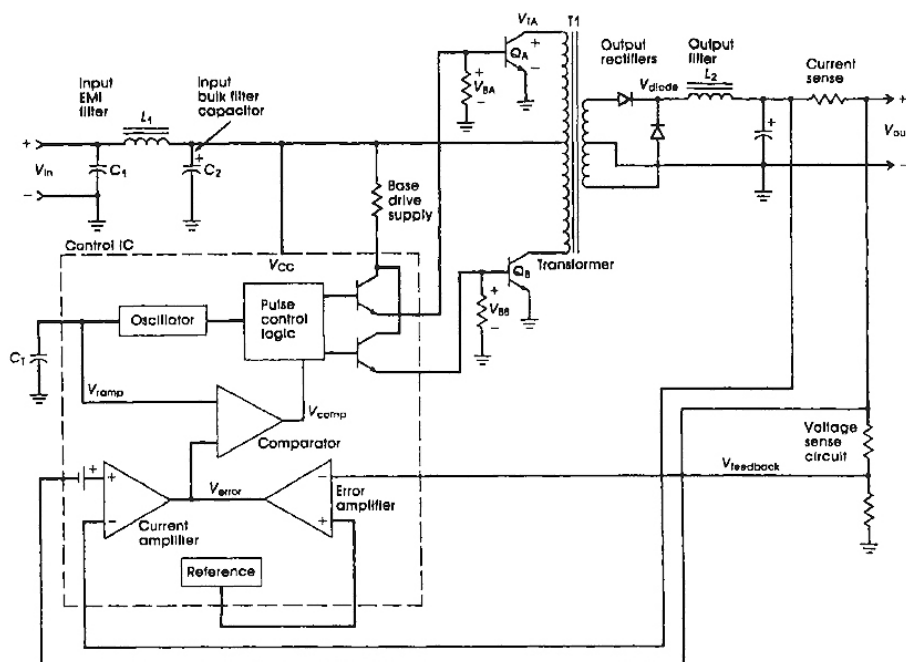
Největší výhodou spínaných zdrojů je právě vysoká účinnost, která se běžně pohybuje v rozmezí 68-90 %. U pulzních zdrojů pracují spínací prvky ve spínacím režimu, pracovním režimu, kde je jejich účinnost největší. Díky tomu může být jejich ztrátový výkon řádově menší než příkon zátěže.

Základním principem je převedení vstupního stejnosměrného proudu na střídavý pomocí spínacího prvku a následná transformace na magnetickém prvku, následkem čeho lze snadno přidávat možná výstupní napětí, stačí přidat další vinutí. V případě transformace napětí je také výhodou nezávislost výstupního napětí na vstupním. Pracovní frekvence spínaných měničů se běžně pohybují nad akustickým pásmem, tedy na frekvenci větší než 20 kHz. Kvůli tomu nejsou potřeba velké a těžké transformátory a velké kapacity, což se pozitivně odráží na ceně.

Základní nevýhodou pulzních měničů je složitější konstrukce a s tím spojený složitější návrh. Tím, že pulzní zdroj rozděluje vstupní napětí na časově omezené impulzy, je omezena reakční doba na změnu zatížení zdroje, tento jev je nazýván přechodná doba odezvy. Toto chování je nutno ošetřit zvětšením kapacit ve výstupním filtru, které kompenzují pokles napětí po dobu, než spínaný zdroj stihne zareagovat.

2.2 Části pulzních měničů

Ukázkové schéma pulzního měniče je uvedeno na obrázku 1. Následující část textu se bude odkazovat na tento obrázek.



Obrázek 1: Schéma vzorového měniče [3].

2.2.1 EMI filtr

Název je odvozen z anglického Electromagnetic interference - tedy filtr elektromagnetického rušení. Tato část se skládá z L-C filtru, který je zařazen mezi zdrojem a samotným měničem. Účelem tohoto filtru je omezit vysokofrekvenční rušení, které se může ze spínaného měniče vracet zpátky do zdroje. Dolní mezní frekvence filtru by neměla být 2 až 3 krát větší než pracovní frekvence samotného měniče.

Dalším účelem filtru je nutnost přidat před vstupní kondenzátor impedanci v podobě indukčnosti, která je schopna omezit špičky přicházející ze sítě.

2.2.2 Vstupní kapacita

Tento prvek je velkou položkou v ceně měniče. Je zodpovědný za uložení dostatečného množství vysoko i nízko frekvenční energie pro spínací cyklus výkonových prvků. Je třeba jej složit minimálně ze dvou částí. Elektrolytického nebo tantalového kondenzátoru pro pokrytí spínací frekvence - nízkofrekvenční proudy a keramického pro pokrytí vyšších harmonických frekvencí - vysokofrekvenční proudy. Tato část také brání vlivu dlouhých přívodů, které se chovají v obvodu jako sériový odpor a indukčnost omezující schopnost zdroje dodat měniči vysokofrekvenční proud, který je potřebný pro rychlou odezvu měniče na změnu zatížení.

2.2.3 Transformátor

Poskytuje stejnosměrné oddělení vstupu a výstupu. Je také určen ke změně velikosti napětí a to ať směrem nahoru nebo dolů. V některých topologiích může být použit i k uložení energie. Výhodou je možnost použití více sekundárních vinutí, spínaný zdroj může tedy dodávat více různých napětí, což například známe u počítačových zdrojů. Správný návrh transformátoru je velmi důležitý a může velice ovlivnit provozní charakteristiky měniče.

2.2.4 Spínací prvky

Jako spínací prvky se používají výkonové bipolární tranzistory nebo MOSFET tranzistory, které dokáží velmi rychle přejít z uzavřeného stavu do stavu plné saturace. Otevírání tranzistorů je řízeno řídicí jednotkou, která vyhodnocuje požadavek zátěže na dodávanou energii. Koncové tranzistory představují jednu z nejvíce namáhaných součástí měniče.

2.2.5 Výstupní usměrňovač

Slouží k usměrnění napětí transformátoru před zátěží. Často se používá topologie dvoucestného usměrnění u transformátoru s vyvedeným středem. Usměrňovač musí být dimenzován na maximální špičku proudu, ne na hodnotu střední.

2.2.6 Výstupní filtr

V našem ukázkovém schématu je použit L-C článek. Je určen k uložení energie pro zátěž po dobu kdy není dodávána měničem. Většinou je tento filtr předdimenzován přibližně o 50 %, aby bylo možné pokrýt náhle změny odběru zátěže do doby, než měnič dodá potřebnou energii.

2.2.7 Proudová sonda

Proudová sonda snímá výstupní proud měniče převodem na napětí. Informace o výstupním proudu je částí regulační smyčky. Hlavním smyslem je ochrana proti přetížení měniče. Řídicí elektronika může implementovat různé druhy omezení proudu, umožňuje například použití měniče jako zdroje konstantního proudu, kdy proudová sonda je po celou dobu součástí regulační smyčky. Pokud se měnič chová jako zdroj napětí, je použita většinou pouze ochrana proti nad-proudu a následnému přetížení měniče.

2.2.8 Napěťová zpětná vazba

Napěťový dělič na výstupu měniče zmenšuje napětí, které je potom použito ve zpětné vazbě pro snímání chybové hodnoty, tedy rozdílu mezi požadovanou hodnotou a skutečnou hodnotou a dále vyhodnocováno pro určení doby plnění.

2.2.9 Řídící jednotka

Tato část je většinou postavena kolem monolitického integrovaného obvodu, který plní veškeré potřebné funkce. V určitých případech můžou být některé části z IO vyčleněny, například u měničů s příkonem nad 2 W je vhodné použít externí spínací prvky. Mezi základní funkce tohoto bloku patří:

- Snímání a korekce výstupního napětí
- Kódování PWM
- Zdroj referenčního napětí
- Generátor nosného kmitočtu
- Ochrana proti nad-proudu a přetížení
- Obvody budiče koncových prvků, případně můžou být integrovány přímo koncové prvky

Další možné části, které nejsou pro funkci měniče přímo potřebné:

- Obvody plynulého náběhu - anglicky nazývané softstart
- Externí synchronizace
- Vzdálené řízení
- Komunikační obvody

2.2.10 Korekce výstupního napětí

Chybový zesilovač zesiluje rozdíl mezi ideálním referenčním napětím a snímaným napětím ze zpětné vazby, ten je pak zpracován pomocí kódování PWM. Modulátor na změnu chybového napětí reaguje příslušnou změnou doby plnění.

2.2.11 Budiče

Koncové spínací prvky je nutno správně budit, jak v případě použití bipolárních tranzistorů tak MOSFETů. U tranzistorů MOSFET nestačí jen podmínka dodání dostatečného napětí, ale zvláště dostatečné rychlé vybití a nabití kapacity C_g .

2.3 Topologie měničů

Spínané zdroje získaly velkou popularitu s příchodem prvních bipolárních tranzistorů ačkoli základní principy funkce jsou známy už z třicátých let dvacátého století. Pulzní měniče vyhovují pro mnoho aplikací, jejich velká oblíbenost měla vliv na vývoj mnoha různých variant. Každá z nich je vhodná pro jiné případy užití. Různých typů topologií existují desítky, dále budou diskutovány jen ty, které je možné využít k naplnění cílů práce.

2.3.1 Kritéria volby topologie

Před začátkem vývoje je nutno porozumět a zhodnotit několik kritérií, které přímo ovlivňují volbu topologie.

1. Špičkový proud primární části, který přímo ovlivňuje stressové namáhání komponent
2. Napětí a tvrdost jeho zdroje
3. Jak velkou část B-H charakteristiky můžeme využít během každého cyklu
4. Galvanické oddělení výstupu
5. Cena a složitost výsledného měniče

Protože je cílem měnič, který bude mít na výstupu efektivní hodnotu napětí 230 V, primární strana bude napájena z baterie a jeho výstupní průběh bude co nejvíce blízký sinusovému, budou dále diskutovány jen ty topologie, které jsou schopny toho dosáhnout. Pro toto použití je vhodná topologie s transformátorem.

Nevhodnou topologií je jednočinný měnič, protože neumožňuje dosáhnout na sekundární straně vhodného průběhu a při použití napětového mezistupně je vhodný jen pro malé příkony.

Mezi vhodné topologie lze zařadit dvojčinný měnič a to ve třech možných konfiguracích:

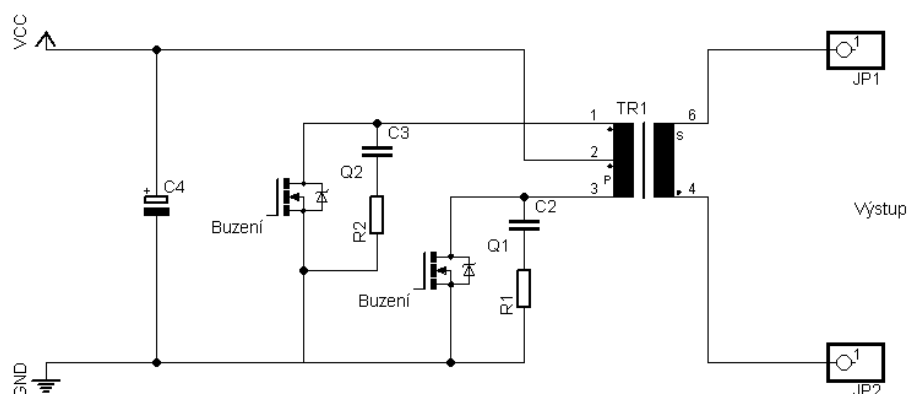
- Paralelní dvojčinný (push-pull)
- Polomůstek (half-bridge)
- Můstek (H-bridge)

2.3.2 Paralelní dvojčinný

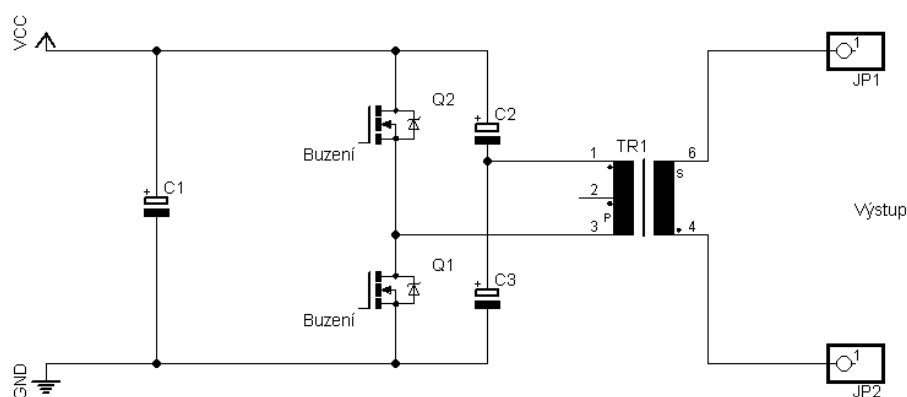
Tato topologie obsahuje dva spínací prvky 2, které nepracují současně ale střídají se, proto dvojčinný. Primární vinutí transformátoru je vinuto ve stejném směru se středem připojeným k napájecímu napětí. Proud polovinami vinutí ale protéká v opačném směru a mění tím směr magnetického toku. Díky tomu je využívána celá hysterezní smyčka materiálu jádra, což má pozitivní vliv na účinnost a tím i potřebnou velikost transformátoru. Tyto vlastnosti určují tuto topologii jako vhodnou pro měniče s příkonem v řádech stovek wattů.

U měniče využívajícího tuto topologii, je kriticky nutné, aby doba plnění nikdy nedosáhla 100 %, protože spínacím prvkům trvá 100 ns - 10 μ s než dosáhnou plného uzavření. Došlo by k magnetickému zkratu uvnitř transformátoru následovaného nárůstem proudu a s velkou pravděpodobností zničením spínacích prvků.

Při sepnutí se na protilehlém spínacím prvku objeví dvojnásobek napájecího napětí, spínací prvky na to musí být dimenzovány.



Obrázek 2: Topologie paralelní dvojčinný (push-pull)

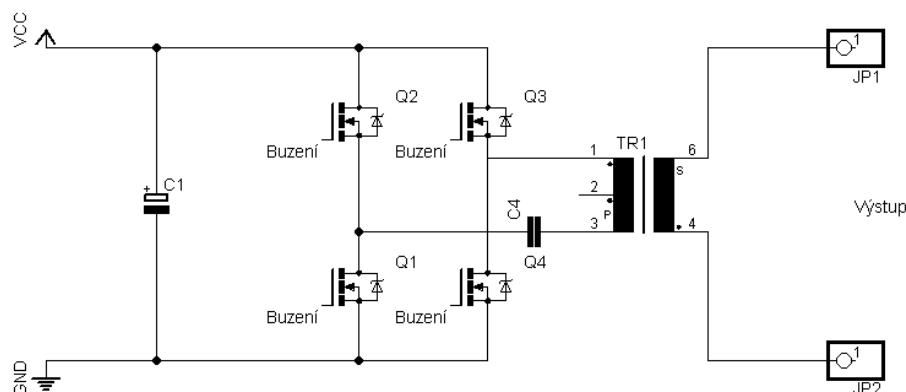


Obrázek 3: Polomůstek (half-bridge)

Nevýhodou topologie je, že neexistují dva shodné tranzistory ani dvě stejné vinutí transformátoru. Následkem je, že jedna polovina vinutí má větší špičkový proud a je blíže saturaci jádra. Při prudké změně zatížení může dojít k tomu, že jedna ze polovin dosáhne saturace. Toto chování se nazývá nestabilita jádra a může mít za následek poškození měniče [3, strana 37]. Pro ochranu před tímto jevem je vhodné řídicí elektroniku doplnit obvodem pro hlídání proudu s velmi krátkou dobou reakce.

2.3.3 Můstek a polo-můstek

Topologie polo-můstku je zobrazena na obrázku 3 a můstku na 4. Tyto topologie mají jedno primární vinutí. U polo-můstku je jeden konec připojený mezi pár tranzistorů, jeden připojuje napájecí napětí a druhý nulový potenciál. Druhá strana vinutí je připojena mezi pár sériově řazených kondenzátorů. Napětí středu spojení kondenzátorů je přibližně polovina napájecího napětí. To znamená, že při provozu se na primárním vinutí vyskytuje maximálně $1/2$ napájecího napětí.



Obrázek 4: Polomůstek (full-bridge)

Na rozdíl od paralelního dvojčinného zdroje netrpí nestabilitou jádra, napětí kapacitního děliče se přizpůsobí rozdílnému magnetickému toku v jádře.

Můstek využívá dva páry spínacích prvků, uspořádané do tvaru písmene H, spínají vždy v protilehlých dvojicích. Maximální napětí na vinutí transformátoru odpovídá napájecímu. Díky nahrazení kapacitního děliče dalšími tranzistory, dosahuje toto uspořádání mnohem lepší účinnosti a je vhodné pro vysoké výkony do několika kilowatt příkonu.

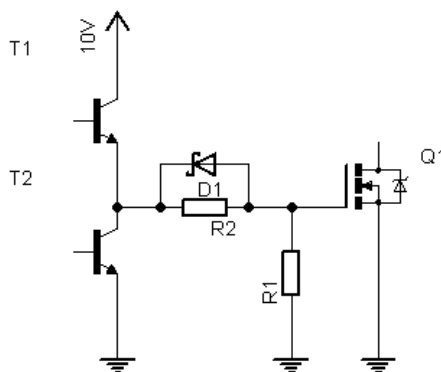
Stability jádra se dosahuje zařazením malého kondenzátoru sériově s vinutím (C4 na obrázku 4). Stejnoseměrné napětí na kondenzátoru se mění podle magnetického toku jádra.

2.3.4 Výkonové MOSFET tranzistory

V minulosti byly používány i bipolární tranzistory. Dnes se díky rozvoji technologie MOSFET používají výhradně tyto unipolární tranzistory. Jejich výhodou je velmi velká rychlost, která je desetkrát větší než u bipolárních tranzistorů a velmi malý odpor ($R_{DS(on)}$) v sepnutém stavu. MOSFETy jsou napětím řízené s izolovanou řídicí elektrodou. Plné otevření tranzistoru nastává přibližně při 10 V řídicího napětí.

2.3.4.1 Důležité parametry Tyto parametry jsou určující při výběru vhodného typu.

- I_{DSS} Maximální proud, který může téci tranzistorem
- U_{DS} Maximální možné napětí mezi elektrodou drain a source. Nesmí být nikdy ani chvilkově překročeno.
- P_D Maximální ztrátový výkon na tranzistoru
- $R_{DS(on)}$ Stejnoseměrný odpor v sepnutém stavu
- Přítomnost ochranné diody



Obrázek 5: Budič typu totem

2.3.4.2 Buzení Na řídicí elektrodu můžeme pohlížet jako kapacitu připojenou mezi gate a source elektrodu (C_{GS}). Rychlé spínání a vypínání vyžaduje velké proudy až několik ampérů pro nabíjení a vybíjení této kapacity, to klade velké nároky na buzení. Například impedance budiče nesmí překročit 200 Ω . Správně dimenzovaný budič umožňuje dosáhnout spínacích časů 30-50 ns.

Nejčastěji používanou topologii budiče je tzv. totem. Poskytuje nízko-impedanční plně aktivní výstup. Tranzistor T1 na obrázku 5 přivádí napájecí napětí na řídicí elektrodu MOSFET tranzistoru. T2 slouží k odvedení náboje z C_{GS} do země. Rezistor R1 slouží k upravení impedance budiče na požadovanou úroveň. Na obrázku je také doplněna vypínací dioda D1, která upravuje impedanci pro uzavírání MOSFETu a tím zkracuje dobu uzavření.

2.3.4.3 Ochranná dioda Při spínání indukční zátěže, například vinutí transformátoru, dochází při vypínání k indukci napětových špiček opačné polarity. Dioda chrání tranzistor před těmito špičkami.

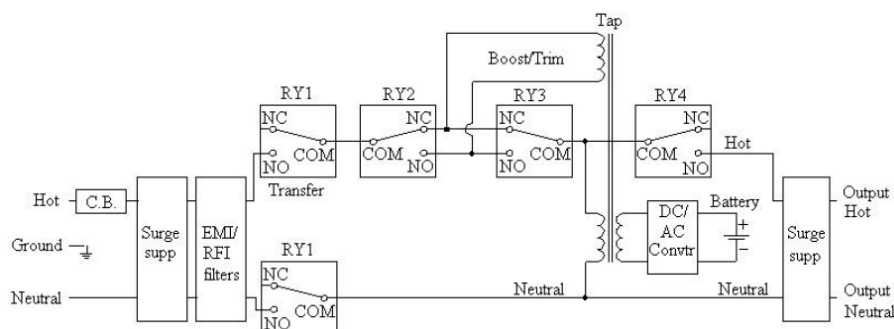
2.4 Řešerše obdobných zapojení

Měniče pro účely přeměny napětí baterie na síťové napětí jsou velmi podobné UPS, proto jsou právě schémata UPS dobrým zdrojem informací k tomuto tématu. Dalším informačním zdrojem, který je dostupný, jsou vzorové návrhy UPS výrobců elektronických součástek.

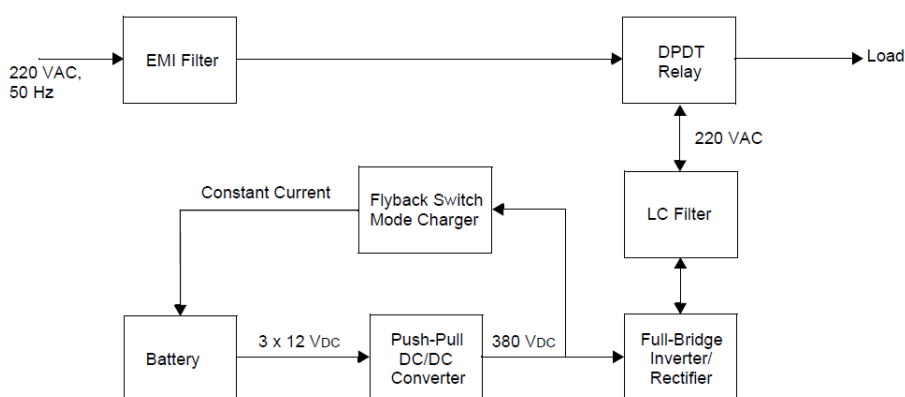
2.4.1 APC SmartUPS 620

Jedná se o zástupce UPS zcela klasické a osvědčené koncepce s transformátorem.

Výstupním signálem této UPS je 50 Hz PWM sinus s mezikrokem. Konstrukce měniče je dvojčinná s transformátorem s jádrem ze železných plechů. Jako řídicí obvod je použit proprietární analogový obvod firmy APC. UPS disponuje ochranou proti přepětí



Obrázek 6: Blokové schéma APC SmartUPS



Obrázek 7: Topologie UPS Microchip

a podpětí na vstupu. Při provozu ze sítě je výstupní transformátor využíván jako regulační autotransformátor 6 a pomocí přepínání vinutí je voleno výstupní napětí. Spínacích prvků je použito 8 ks tranzistorů MOSFET. UPS je dimenzována na zátěž 390 W nebo 620 VA. Komunikace UPS s PC je řešena pomocí USB přes dodatečný integrovaný obvod.

2.4.2 Microchip - Digital Pure Sine Wave UPS Design Package

Referenční koncept firmy Microchip s návrhem implementace hardwarové části a softwaru při použití mikrokontroléru dsPIC. dsPIC spadá do kategorie plně 16-bitových až 32-bitových mikrokontrolérů s výkonem až 70 MIPS a podporou instrukcí, které umožňují pracovat jako digitální signálový procesor. Speciálně pro aplikace buzení motoru nebo práci v měničích jsou určeny modely s až 30 PWM výstupy a hardwarovou násobičkou 16x16 s 40 bitovým výstupním akumulátorem.

Firma Microchip nabízí volně ke stažení kompletní návrhy celého řešení včetně několika možných řešení jednotlivých částí a kompletního softwaru pro řídicí část.

Tato UPS je navržena jako tzv. offline 7. V případě, že je dostupné síťové napětí, tak zátěž je napájena z něj. UPS je v pohotovostním stavu a dochází jen k nabíjení baterie. Při

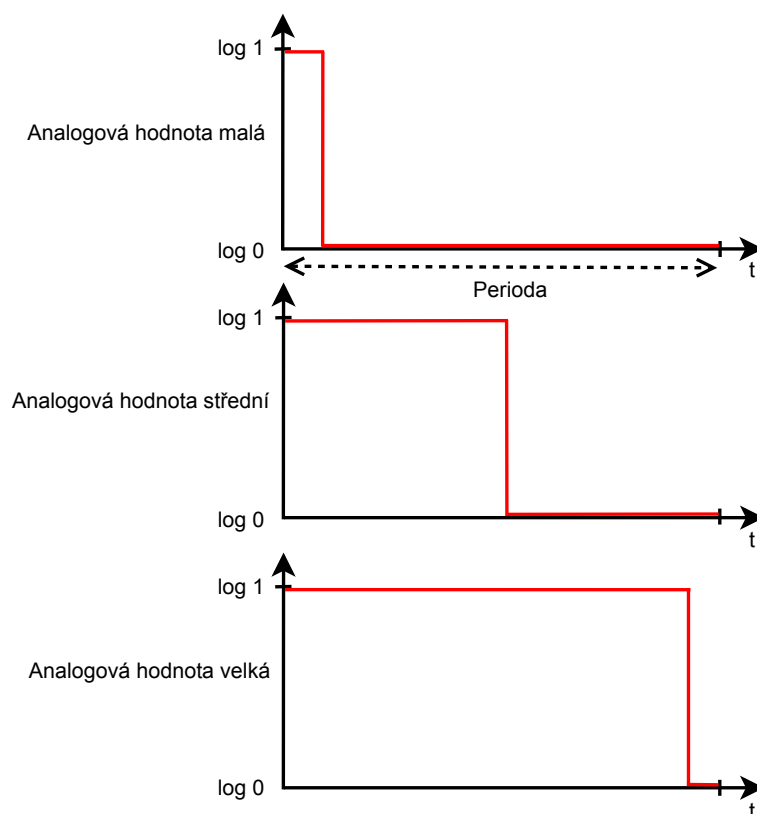
výpadku dojde k přepnutí relé na měnič, který převádí napětí baterie na napětí potřebné pro spotřebič. Nevýhodou je doba detekce výpadku a následného přepnutí. U tzv. online UPS je do cesty zařazen mezistupeň, do kterého může být dodávána energie ze sítě nebo z baterie. Z mezistupně je napájen měnič pro připojený spotřebič. Toto uspořádání má nevýhodu v celkově nižší účinnosti než u offline verze.

Celá konstrukce se skládá ze tří měničů. První část tvoří můstek, který při provozu ze sítě slouží i jako usměrňovač pro napětový mezistupeň, jehož napětí je voleno jako amplituda síťového. Při provozu z baterie zde dochází k modulování sinusového průběhu pomocí PWM. Při provozu z baterie je proud dodáván do mezistupně skrz DC/DC paralelní dvojčinný měnič s transformátorem. Zde také dochází k regulaci a snímání proudu a napětí tak, aby napětí mezistupně bylo konstantní. V návrhu je primární strana opatřena snímáním proudu pro potřeby ochrany před saturací jádra transformátoru. Poslední částí je jednočinný zdroj pro nabíjení akumulátoru konstantním proudem.

Softwarová část využívá všech výhod, které může takto výkoný mikrokontrolér poskytnout. Je využito na 8 analogových vstupů se vzorkovací frekvencí minimálně 25 kHz. Měření všech veličin je prováděno plně digitálně včetně měření efektivních hodnot.

3 Pulzně šířková modulace

Pulzně šířková modulace je druh diskrétní nekvantované modulace určené pro přenos analogových signálů pomocí signálu dvoustavového. Označení PWM je odvozeno od anglického názvu Pulse Width Modulation, v češtině je občas používána zkratka PŠM, která vznikla z názvu pulzně šířková modulace. Hodnotovou veličinou je nejčastěji napětí, ale může být i proud nebo světelný tok. Informace o analogové veličině je přenášeno pomocí šířky impulsu.



Obrázek 8: Příklad závislosti střídy na hodnotě vstupního napětí

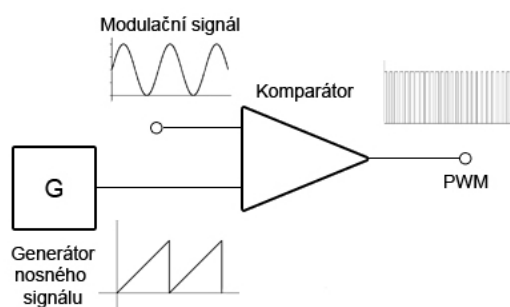
3.1 Principy PWM

Modulace je dvoustavová. Modulovaný signál nabývá dvou logických hodnot log. 0 a log. 1. Analogová informace je kódována jako střída signálu s pevnou periodou signálu. Pokud modulační signál nabude hodnoty 0 %, tak nedojde k vygenerování impulsu o hodnotě log. 1. Při vstupním signálu o hodnotě 50 % bude mít logická 1 délku 50 % periody signálu. Pro 100 % dojde k naplnění celé periody logickou hodnotou 1. Příklad vztahu mezi hodnotou vstupního napětí a šířky impulsu je uveden v obrázku 3. Pro poměr mezi

dobou logické 1 a 0 se používá výraz střída, pro dobu trvání impulsu lze použít název doba plnění.

Z tohoto principu je zřejmá i jedna z nevýhod PWM a to nutnost znát při návrhu modulátoru možný rozsah modulačního napětí, protože při překročení předem stanovené hodnoty 0 % směrem dolů a 100 % směrem nahoru dojde ke ztrátě a zkreslení informace. Například frekvenční modulace touto nevýhodou netrpí.

Pro správnou funkci PWM je nutno splnit Shannonův-Nyquistův-Kotělnikovův teorém. To znamená, že nosná frekvence musí být nejméně dvakrát větší než frekvence modulačního signálu.



Obrázek 9: Blokové schéma modulátoru

3.2 Způsob modulace a demodulace

Blok modulátoru lze principiálně složit ze dvou jednoduchých částí. Z generátoru nosného kmitočtu a komparátoru. Blokové schéma takového jednoduchého modulátoru je znázorněno na obrázku 3.1.

Nosný kmitočet má pilový nebo trojúhelníkový tvar. Komparátor porovnává napětí modulačního a nosného signálu. Pokud je velikost napětí nosného signálu menší než velikost modulačního, je výstupní hodnotou komparátoru logická 1, v opačném případě je na výstupu log. 0. Převáděný signál má řádově menší kmitočet, během jedné periody nosného kmitočtu je změna napětí modulačního velmi malá. V průběhu periody nosného kmitočtu napětí pily postupně lineárně roste a doba za, jakou dosáhne napětí modulačního, je dobou plnění periody. Dojde tak k odebrání jednoho vzorku.

Velkou výhodou PWM je snadná demodulace. K převedení zpět na původní signál stačí jednoduchá dolní propust. Nejčastěji se používá RC článek druhého řádu.

PWM je v elektronice široce používána. Ve výkonových aplikacích se PWM používá k regulaci veličin. Kde se s výhodou využívá toho, že polovodičové prvky pracují ve spínacím režimu, díky čemuž lze dosáhnout vysoké účinnosti. Další možnosti použití jsou měniče napětí s přesně tvarovaným výstupním napětím.

Pro audio aplikace se používají koncové zesilovače ve třídě D, kde koncový stupeň používá právě pulzní šířkovou modulaci.

4 Regulace

Jak bylo napsáno v kapitole 2, měniče udržují na výstupu konstantní napětí nebo proud. Proces, při kterém udržujeme fyzikální veličinu na požadované hodnotě se nazývá regulace.

U měniče napětí je právě napětí regulovanou veličinou, které je průběžně snímáno a na základě její hodnoty je nastavována střída. Střída je takzvanou nastavovanou veličinou. Cílem tohoto procesu je stav, kdy skutečná hodnota odpovídá hodnotě požadované. Tyto prvky tvoří regulační okruh, nebo také regulační smyčku. Stabilita nastavení může být ovlivněna rušivou veličinou, což má za následek odchýlení skutečné hodnoty od požadované.

V regulační smyčce se regulovaná veličina snímá a srovnává pomocí komparátoru s požadovanou hodnotou. Rozdíl mezi požadovanou a skutečnou hodnotou je regulační diferencí $e = w - x$, kterou regulační člen převede na nastavovací veličinu y a ta je zároveň vstupní veličinou regulovaného systému. Výstupem regulovaného systému je regulovaná veličina x . Schéma regulace je zobrazeno na obrázku 10.

4.1 Druhy regulace

Při ruční regulaci přebere člověk část regulační smyčky na sebe. Nejčastějším druhem je regulace automatická, při které vše probíhá bez vlivu člověka s výjimkou zadání požadované veličiny.

Můžeme také rozlišovat další dva druhy regulace:

- regulace nastavení na pevnou hodnotu - požadovaná hodnota se v čase nemění
- regulaci nastavení na časově proměnnou hodnotu - požadovaná hodnota se v čase mění

Při časově proměnné regulaci může dojít ke kmitání kolem požadované hodnoty a to v případě, pokud je zpětnovazební přenos ve smyčce moc velký. Skutečná hodnota se tak dostane z režimu časového vleku do režimu kmitání.

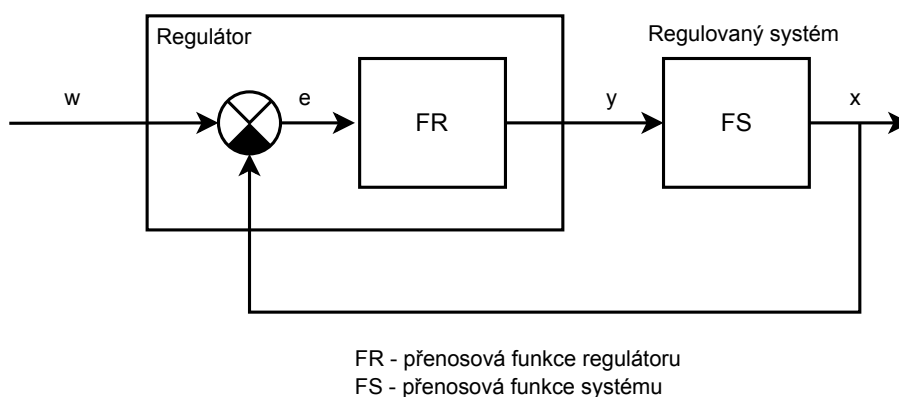
4.2 Regulační členy

Při definování odezvy regulované soustavy na změnu nastavovací veličiny je vhodné použít bloková schémata, která zjednodušují obvod na elementární části - regulační členy, s určitým typem dynamické a statické odezvy. Při popisu celé soustavy jsou tyto regulační členy řetězeny za sebe.

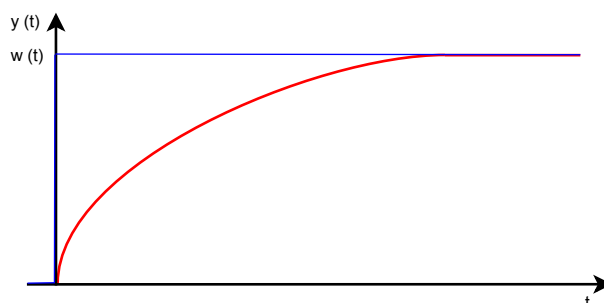
4.2.1 Proporcionální člen

Výstupní signál s_2 je proporcionální vstupnímu signálu s_1 . Funkční závislost výstupu na vstupu je dána násobením konstantním zesilovacím činitelem - konstantou úměrnosti K_P .

$$s_2 = K_P * s_1$$



Obrázek 10: Funkční schéma regulace



Obrázek 11: Odezva PT členu

Odezvou P členu na skokovou funkci je opět skoková funkce. Proto je někdy jako znak P regulátoru používána právě skoková funkce.

4.2.2 Proporcionální člen se zpožděním prvního řádu

Bývá označován jako PT_1 -člen. Výstupní odezva je proporcionální s určitým zpožděním vstupního signálu. Odezva na jednotkový skok je znázorněna na obrázku 11. Odezvou je exponenciální funkce velmi podobná přechodovému jevu při nabíjení kondenzátoru. Přenosová funkce je dána rovnicí:

$$s_2 = K_P * (1 - e^{-\frac{t}{T}})$$

Kde T je časová konstanta a t je čas. Příkladem takové soustavy s touto odezvou může být například motor, který má vždy hmotnost rotoru a trvá nějakou dobu, než dosáhne požadovaných otáček. Odezvou na sinusový signál je opět sinusový signál, s rostoucí frekvencí ale klesá amplituda, protože nabíjení zásobníku energie je stále menší. PT člen tedy vyhlazuje signály vysokého kmitočtu nebo jinými slovy jde o dolní propust. Protože časově zpožďuje vstupní signál dochází k fázovému posuvu. Úhel fázového posuvu se mění od 0 do -90° .

4.2.3 Integrační člen

Je označován jako I-člen. Výstupní signál odpovídá průběhem integrálu funkce vstupního signálu. Tedy integrace narůstající plochy pod křivkou vstupní funkce. Integrační časová konstanta T_I udává čas, za který dosáhne výstupní veličina hodnoty 1. Převrácená hodnota časové konstanty se nazývá integrační koeficient K_I . Odezva na jednotkový skok je dána rovnicí:

$$s_2 = \frac{1}{T_I} * t = K_I * t$$

Odezvou I-členu na sinusový signál je signál cosinusový. Při malých kmitočtech mnohem menších než K_I je výstupní signál mnohem větší než vstupní. Pokud je frekvence vstupního signálu mnohem větší než K_I je odezva mnohem menší než vstupní signál. Fázový posuv je konstantní s hodnotou -90° , což znamená, že výstupní signál je zpožděn za vstupním.

4.2.4 Derivační člen

Člen jehož odezva odpovídá derivaci vstupního signálu je označován jako D-člen. Odezvou takového prvku na jednotkový skok je Diracův impuls. Odezvou na lineárně rostoucí funkci je konstantní funkce.

D-člen se chová jako horní propust, nepropouští nízké frekvence a tedy amplituda výstupního signálu je závislá na frekvenci vstupního signálu. Fázový posuv je konstantní o velikosti 90° , což znamená, že výstupní signál předbíhá vstupní signál.

4.3 Regulátory

Slouží k ovlivňování regulovaného systému automatizovanou regulací a k dosažení a udržování požadovaného stavu. Regulátor zjišťuje stav systému, vyhodnocuje ho a upravuje okamžitou výstupní hodnotu tak, aby odstranil odchylku skutečné hodnoty od požadované.

4.3.1 P regulátor

Výstupní hodnota P regulátoru je úměrná regulační diferencí e , respektive rozdílem mezi požadovanou a skutečnou hodnotou. P regulátory pracují bez zpoždění a s trvalou regulační diferencí, která by mohla být teoreticky odstraněna za nekonečně dlouhou dobu¹.

4.3.2 PI regulátor

Vytváří nastavovací veličinu součtem dvou složek. Jedna složka je vytvořena proporcionální odezvou na regulační diferencí a druhou složku tvoří odezva I-členu, která je

¹Toto tvrzení platí pro analogový regulátor, u diskrétního bude vždy odchylka minimálně ve velikosti jednoho kroku úrovně diskretizace

úměrná součinu P složky a času. Výhodou PI regulátoru je, že vlivem I-členu plně odstraňuje regulační diferenci.

4.3.3 PID regulátor

Je univerzálnější než předchozí dva. Nastavovací signál tvoří vážený součet proporcionální odezvy P regulátoru, integrálu P složky (I-složka) a derivací P složky (D-složka).

Při vzniku regulační difference vytvoří P-člen okamžitou trvalou složku nastavovacího signálu. I člen vytvoří účinnou složku nastavovacího signálu i při malé regulační diferenci, kterou díky tomu může odstranit v dosažitelném čase. D člen se uplatní při změně regulační difference a předvídavým překmitem nastavovacího signálu urychlí vyrovnání vzniklé regulační difference.

U PID regulátoru stoupá složitost nastavení správných hodnot pro optimální regulaci. Z tohoto důvodu jsou často používány P nebo PI regulátory. Volba vhodného regulátoru záleží také na vlastnostech regulovaného objektu. Hlavním požadavkem na volbu regulátoru je výsledná stabilita regulační smyčky.

4.4 Číslicová regulace

Při použití počítače pro účely regulace přebírá počítač úlohu výpočtu regulační difference a podle naprogramovaného regulátoru vypočítává nastavovací veličinu, kterou potom předává aktoru v regulovaném procesu. Také často provádí kontrolu zda regulační difference nepřekračuje meze bezpečnosti a dohlíží tak na bezpečnost regulovaného procesu.

Parametry adaptivního regulačního systému můžou být upravovány změnou parametrů tak, aby byla regulace optimální, rychlá a bez nežádoucího kmitání.

4.4.1 Digitalizace signálu

Počítač není schopen pracovat se spojitými signály, které se v regulovaných obvodech vyskytují. Je nutno analogové signály digitalizovat, tedy vzorkovat a kvantovat. Podle počtu rozlišitelných stupňů je nutno volit správnou délku slova, počet možných hodnot je hodnotovou diskretizací. Numerická rozlišitelnost počítače by měla být větší než rozlišitelnost převodu snímače v regulační smyčce aby bylo možné co nejvíce potlačit vliv chyby způsobené zpracováním dat. Na rozdíl od spojitého regulátoru není počítač schopen pracovat se signály, které jsou v čase spojitě, je nutné je vzorkovat s určitou opakovací periodou. Jedná se o časovou diskretizaci. Čím větší je frekvence vzorkování a počet stupňů převodníku, tím více odpovídá digitální signál původnímu analogovému.

Vzorkování a jejich uchovávání v paměti způsobuje fázové zpoždění signálu o velikosti jedné poloviny vzorkovací frekvence. Toto zpoždění má na regulační smyčku stejný vliv jako fázový posuv a zvyšuje náchylnost ke kmitání.

Stejně jako u analogové regulace vypočítává při číslicové regulaci PID algoritmus² hodnoty nastavovací funkce. Algoritmus je realizován programem. Algoritmy můžeme rozdělit na nastavovací a rychlostní.

²Formální postup s konečným počtem kroků vedoucí k nějakému cíli.

4.4.2 Nastavovací PID algoritmus

Nastavovací algoritmus počítá hodnoty nastavovací funkce z diskretní regulační difference, která je vypočítána z hodnot poskytnutých AD převodníky.

Při číslicové regulaci se difference mění skokově v okamžicích T_A až $n * T_A$ a nabývá hodnot e_1 až e_n . Při výpočtu je integrál nahrazen sčítáním hodnot diskretní funkce a derivace rozdílem sousedních hodnot. Jednotlivé vypočítané složky jsou sčítány. Složky lze vypočítat podle následujících rovnic:

$$\begin{aligned} \text{P-část: } y_{Pn} &= K_p * e_n \\ \text{I-část: } y_{In} &= K_p * \frac{T_A}{T_n} \sum_{i=0}^n e_i \\ \text{D-část: } y_{Dn} &= K_p * \frac{T_V}{T_A} * (e_n - e_{n-1}) \end{aligned}$$

K_p	Konstanta úměrnosti
T_n	Integrační časová konstanta
T_V	Derivační časová konstanta
T_A	Vzorkovací perioda

4.4.3 Rychlostní PID algoritmus

Na rozdíl od nastavovacího algoritmu, který počítá hodnoty nastavovací funkce, tak rychlostní algoritmus počítá jen její přírůstky $\Delta y_n = y_n - y_{n-1}$. Ty pak přičítá k hodnotě nastavovací funkce y_{n-1} , která je uložena v paměti.

Použitím rovnice uvedené v předchozím odstavci a rovnice nastavovacího PID algoritmu lze získat následující rovnici pro rychlostní algoritmus [5]:

$$\Delta y_n = K_P \left[e_n - e_{n-1} + \frac{T_A}{T_I} * e_n + \frac{T_V}{T_A} (e_n - 2e_{n-1} + e_{n-2}) \right] \quad (1)$$

4.5 Nastavení regulátoru

Dobrá regulace je podmíněna správným nastavením regulátoru. Většinou je regulátor nastaven tak, aby skutečná hodnota požadovanou hodnotu dosáhla v co nejmenším čase a pokud možno jí co nejméně překmitla. Rychlost je posuzována podle času regulační odezvy t_0 , za který dosáhne požadované hodnoty.

Jednou z možných metod nastavení je Ziegler–Nicholsova heuristická³ metoda [8], česky bývá nazývána metodou kritického zesílení.

Při nastavování parametrů je prvně vyřazena integrační a derivační složka a konstanta proporcionality K_P je zvětšována od nuly dokud nedosáhne hodnoty K_U , při které se regulační smyčka netlumeně rozkmitá s konstantní amplitudou a dobou kmitu T_p . Parametry proporcionální integrační a derivační složky pak vypočítáme podle tabulky 1.

³Metoda využívající zkusmé řešení problému. Řešení je často přibližné a založené na poučeném odhadu nebo zkušenostech.

Regulátor	K_P	K_I	K_D
P	$0,5 K_P$	-	-
PI	$0,45 K_u$	$\frac{1,2 K_p}{T_p}$	-
PD	$0,8 K_u$	-	$\frac{K_p T_p}{8}$
PID	$0,60 K_u$	$\frac{2 K_p}{T_p}$	$\frac{K_p T_p}{8}$

Tabulka 1: Nastavení PID regulátoru podle Ziegler–Nicholsovy metody

5 Návrh a realizace jednoduchého měniče

Pro účely testování a vývoje firmware bylo nutné navrhnout prototyp měniče s transformátorem, který bude schopen na sekundární straně udržet napětí 230 V.

Návrh měniče je složen z jednotlivých bloků, které byly představeny v předchozí kapitole. Jednotlivé části lze řešit různými způsoby a velmi často je možné jednotlivá řešení vzájemně kombinovat s cílem zlepšit vlastnosti měniče.

V tomto návrhu je kladen důraz na jednoduchost a bezproblémovou funkčnost zvláště pro možnosti testování software. Požadavky na měnič, který bude použit při vývoji firmware:

- Jednoduchá konstrukce
- Napájení z baterie 12 V
- Střídavé výstupní napětí s efektivní hodnotou 230 V
- Příkon do 100 W
- Galvanické oddělení vstupu a výstupu
- Ochrana před přetížením

Pro malé příkony do stovek wattů je vhodná paralelní dvojčinná topologie. Další výhodou této topologie je pro účely takového měniče snadno dostupná součástková základna zvláště transformátorů, tlumivek a jiných, ne zcela běžně v obchodě dostupných součástek.

Baterie je připojena přes ochranné relé, které slouží jako bezpečnostní pojistka například při změně způsobu modulace, také pokud měnič nepracuje je baterie odpojena. Před relé je také připojen vysoko-impedanční dělič napětí z jehož výstupu je odebíráno napětí pro měření napětí baterie. Za kontakty relé je připojen vstupní filtr složený z cívky L_1 a z elektrolytického kondenzátoru C_1 a keramického kondenzátoru C_2 .

Jako spínací prvky jsou použity MOSFET tranzistory s N indukovaným kanálem IRFZ44 jehož parametry jsou:

- $V_{ds} = 60 \text{ V}$
- $R_{DS(on)} = 28 \text{ m}\Omega$
- $I_d = 50 \text{ A}$
- $P_d = 150 \text{ W}$
- Pouzdro TO220
- Ochranná dioda

Tyto parametry jsou dostatečné a vhodné pro dané použití. V případě nutnosti zvětšení výstupního výkonu stačí přidat další pár tranzistoru paralelně ke stávajícím. Tranzistory jsou doplněny RC článkem, který omezuje $\frac{dU}{dt}$, tedy zajistí, že nárůst napětí D-S od nuly do maxima je mnohem pomalejší než než přechod MOSFETu z vodivého do nevodivého stavu. MOSFET je uzavřen dříve než napětí D-S vzroste na zlomek maxima [3, kapitola 11].

Budiče koncových prvků jsou postaveny z diskrétních součástek a využívají totem topologii složenou z komplementárních bipolárních tranzistoru KD139 a KD140.

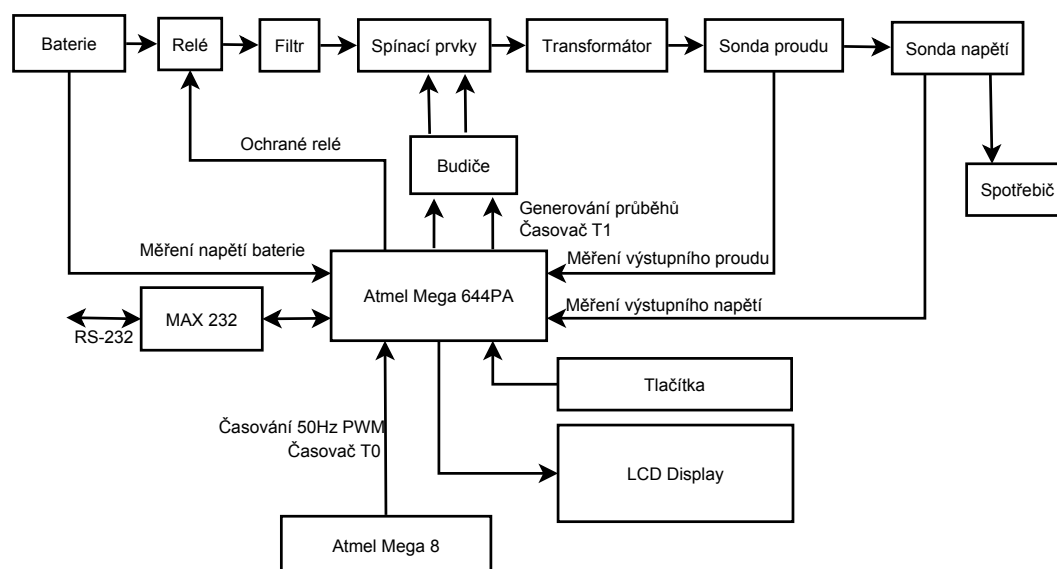
Transformátor je použit z vysloužilého kusu APC SmartUPS 620, stejně ostatní použité kusy indukčností. Na sekundární straně výstupního transformátoru jsou zařazeny prvky pro snímání výstupního proudu a napětí. Snímání proudu je provedeno pomocí snímacího odporu a následným galvanickým oddělením optočlenem SHF-615. Měření napětí je realizováno měřícím transformátorem a napětovým děličem.

Funkci řídicího prvku obstarává mikrokontrolér Atmel Mega 644PA s pracovním kmitočtem 16MHz.

Komunikační vývody sériového portu používají napěťové úrovně TTL, které jsou neslučitelné s úrovněmi používaných na rozhraní RS-232. Pro přizpůsobení úrovní je použit integrovaný obvod MAX232, který má integrován měnič díky čemu nepotřebuje kromě napájení 5V další pomocná napětí a vyžaduje pouze dva pomocné kondenzátory. MAX232 plně dvou kanálový, v návrhu je využit pouze jeden.

Pro případ, kdy uživatel nebude mít k dispozici rozhraní RS-232, je měnič vybaven displayem. Display je připojen pomocí čtyř datových vodičů a tří řídicích. V návrhu je použit znakový display DEM16101SYH-LY firmy Display Elektronik, který je založen na čipu Hitachi HD44780. Pro ovládání jsou připojeny k MCU tři mikrosvínače.

Spolupráce jednotlivých částí měniče s mikrokontrolérem je znázorněna blokovým schématem na obrázku 12, elektrotechnické podrobné schéma je součástí přílohy A.



Obrázek 12: Blokové schéma měniče

6 Implementace firmware

Ačkoli je implementace a hardwaru a softwaru oddělen do dvou kapitol, tak při vývoji je nelze zcela oddělit. Implementace firmwaru probíhala souběžně s vývojem hardwaru a s jeho testování. Některá softwarová řešení jsou vyžádána hardwarem a naopak, například obvody měření stavu musí být přizpůsobeny schopnostem AD převodníku.

6.1 Atmel Mega644PA

Celý řídící firmware je navržen pro 8 bitový mikrokontrolér Mega 644PA firmy Atmel. Je založen na RISC jádře harvardské architektury AVR a většinu instrukcí dokáže vykonat v průběhu jednoho cyklu. Dosahuje výkonu 1 MIPS na 1 MHz taktovacích hodin, je osazen 64 kB programové paměti, 2 kB EEPROM a 4 kB RAM paměti.

Jádro AVR je přímo napojeno na 32x8 univerzálních registrů, které mohou být přímo přístupovány aritmeticko-logickou jednotkou (ALU). Kromě ALU je na čipu dostupná i násobička, ta zvládne pomocí specializované instrukce vynásobit dvě 8 bitové čísla za dobu dvou strojových cyklů.

Vývoj programů pro AVR je možný v jazyce C nebo assembleru. Tvůrci jádra AVR se aktivně účastnili vývoje kompilátoru jazyka C, díky tomu jsou výsledné strojové kódy vzniklé překladem dobře optimalizované, efektivní a C je vhodné i pro vývoj výkonově náročnějších aplikací.

6.1.1 Periférie

Mega644 nabízí široké spektrum periférií:

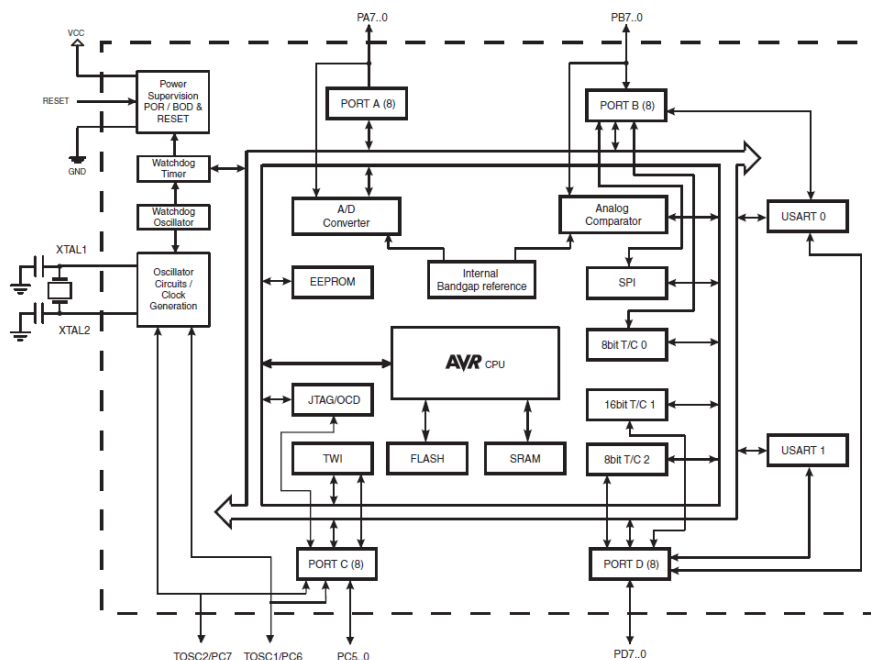
- 4 vstupně-výstupní 8 bitové brány
- Dva 8 bitové a jeden 16 bitový čítač
- 6 kanálů PWM
- Dvě univerzální sériová rozhraní
- 8 kanálový ADC s rozlišením 10 bitů

Blokové schéma zobrazující provázanost jednotlivých periférií a jádra je uvedeno na obrázku 13.

6.1.2 Hodnocení výkonu

Firmware pro měnič obsahuje kritické části, které je nutné vždy vykonat. Mezi ně patří například modulování PWM pro výstup. Na této funkci lze vhodně demonstrovat otázky týkající se výkonu.

Požadavkem je, aby pracovní frekvence měniče ležela nad akustickým pásmem. Při použití 8 bitového čítače v režimu Fast PWM bez děliče hodin a 16 MHz pracovní frekvenci bude výsledná frekvence dána vztahem 2.



Obrázek 13: Blokové schéma mikrokontroléru Mega 644

$$f_{PWM} = \frac{f_{osc}}{prescaler * counter_{max}} = \frac{16000000Hz}{1 * 255} = 62,7kHz \quad (2)$$

Při použití režimu fázově správné PWM čítač počítá do maxima dvakrát. Výsledná frekvence bude dle rovnice 3.

$$f_{PWM} = \frac{f_{osc}}{prescaler * counter_{max} * 2} = \frac{16000000Hz}{1 * 255 * 2} = 31,3kHz \quad (3)$$

Je zřejmé, že u první varianty bude muset jádro obsloužit přerušení od čítače každých 255 strojových cyklů u druhé každých 510 cyklů. Samotné vyvolání přerušení spotřebuje průměrně pět strojových cyklů. Během této doby je aktuální čítač instrukcí uložen na zásobník. Pokud přerušení přijde v době vykonávání instrukce, která vyžaduje více strojových cyklů, dojde prvně k jejímu dokončení. Samotný návrat z přerušení zpět do hlavního programu vyžaduje opět pět strojových cyklů.

Pro první variantu nám tedy zbývá 245 strojových cyklů během kterých musí proběhnout potřebné výpočty. Například výpočet potřebné doby plnění pro PWM a případně obsloužit další vektory přerušení nebo rutiny pro méně důležité části, například komunikaci s PC nebo uživatelské rozhraní.

U kritických částí je vhodné věnovat velkou pozornost výkonové optimalizaci, vhodnou demonstrativní ukázkou je operace násobení a dělení. V tabulce číslo 2 je uveden potřebný počet hodinových cyklů pro vykonání s jednotlivými datovými typy pro vý-

	Strojových cyklů	Velikost kódu
$8 * 8 = 16$ bit bez znaménka	34	34
$16 * 16 = 32$ bit bez znaménka	105	105
$8 / 8 = 8 + 8$ bit bez znaménka	66	58
$16 / 16 = 16 + 16$ bez znaménka	196	173

Tabulka 2: Počet strojových cyklů potřebných pro provedení operace a výsledná velikost programu [2].

počty s celými čísly. Součin čísel datových typů double může vyžadovat až 1800 strojových cyklů.

Je tedy zřejmé, že je vhodné se co nejvíce vyvarovat případným výpočtům a máme-li k dispozici dostatečně velkou paměť řešit tyto operace například aproximací tabulkou (LUT).

6.2 Části firmware

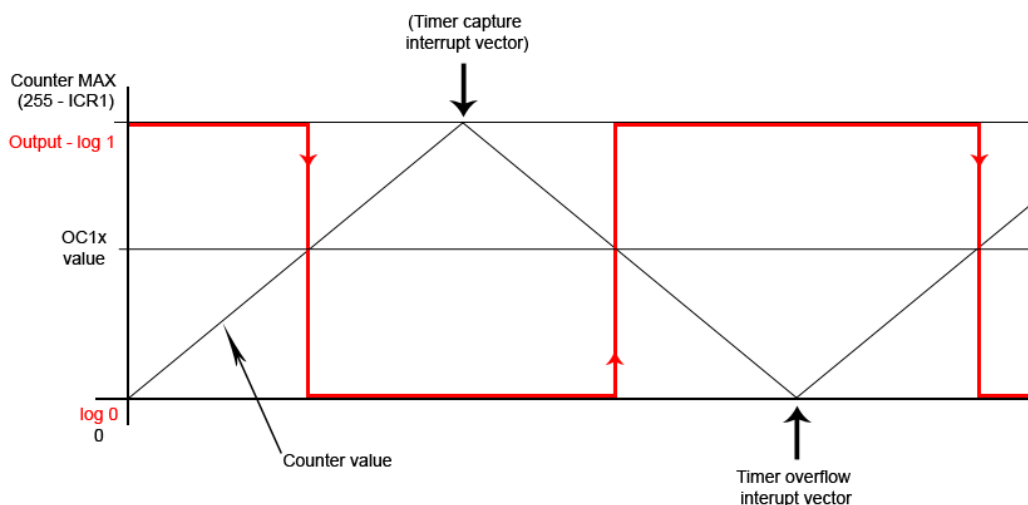
Celý firmware můžeme složit z jednotlivých částí, bloků, které obstarávají jednotlivé části periferii a obsluhovaných funkcí. Pro návrh lze definovat tři kategorie priorit pro obsluhu jednotlivých částí:

- **Vysoká priorita** Tyto části jsou kritické pro chod měniče a musí být vždy obslouženy. Tyto části je vhodné obsluhovat pomocí přerušení. *Generování PWM.*
- **Střední priorita** Části týkající se například komunikace s PC není kriticky nutné obsloužit a lze je obsloužit přerušením. *Měření provozních veličin, komunikace s PC*
- **Nízká priorita** Některé hardwarové nebo softwarové požadavky není nutné obsloužit, nemají vliv na základní funkci firmware. Obsluha částí s nízkou prioritou může být provedena v hlavní programové smyčce. *Obsluha displaye, obsluha vstupních tlačítek.*

Využívání jednotlivých přerušení pro dané aplikace je součástí přílohy D. MCU Atmel Mega nemá podporu priorit přerušení, což je jednou z nevýhod mikrokontroléru této kategorie.

6.3 Inicializace výstupu

Protože budič koncových prvků pracuje v inverzním režimu je nutné aby před spuštěním modulace byla na výstupních portech log. 1. Ochranné relé je ve výchozí poloze vypnuto. První věcí provádějící se po spuštění je inicializace výstupních portů, které budou použity pro buzení tranzistorů a spínání vstupního relé. K jednotlivým portům je přístupováno pomocí maker. Výstupní brána použitá pro buzení je označena jako *DRIVER_PORT* a jednotlivé bity jako *DRIVER_OUTPUT_A* a *DRIVER_OUTPUT_B*, ochranné relé je přístupné pod makry *PROTECT_PORT* a *PROTECT_OUT*.



Obrázek 14: Způsob generování PWM pomocí čítače a využívání registrů

Inicializace je provedena v metodě *initOutput()*, prvně jsou nastaveny požadované bity data direction registrů na hodnotu log. 1 a poté nastaveny samotné porty.

6.4 Modulace sinu pomocí vysokofrekvenční PWM

Pro generování modulovaného sinu pomocí PWM by postačoval 8 bitový čítač, ale generování 50 Hz PWM už vyžaduje použití 16 bitového čítače, protože by bylo nutné snížit pracovní frekvenci MCU tak, aby s použitím děliče kmitočtu čítače bylo možné dosáhnout frekvence výstupu 100 Hz (jednotlivé poloviny vlny 50 Hz). Toto řešení by vedlo k snížení výkonu mikrokontroléru a nejspíše i ke zmenšení přesnosti regulace. Celé řešení modulace PWM se však snaží zachovat minimální rozlišení 8 bitů. Proto je pro oba módy použit časovač 16 bitový.

Pro generování PWM je tedy použit časovač číslo 1, který je 16 bitový a nabízí přesné hardwarové generování fázově správné PWM s proměnou frekvencí.

Časovač je nastaven do režimu Phase Correct Fast PWM 8 bit pomocí WGM bitů v registru *TCCR1A*. Čítač počítá směrem nahoru a poté směrem dolů. Není předřazen žádný dělič kmitočtu, výsledná frekvence je vypočtena v rovnici 3.

Čítač přímo ovládá vstup-výstupní porty a pracuje v invertujícím režimu. Pokud hodnota v registru *OC1A* odpovídá hodnotě čítače při počítání směrem nahoru dojde k přepnutí výstupu *OC1A* do stavu log. 0, při počítání směrem dolů dojde k přepnutí do stavu log 1. Jde o řízení doby logické nuly.

Obsluha využívá jen přerušení při přetečení čítače, což znamená, že byla vygenerována jedna perioda PWM, v tomto okamžiku je nutno dodat informaci o potřebném plnění v další periodě. Způsob generování, využívaná přerušení a hodnoty jednotlivých registrů jsou znázorněny na obrázku 14.

Potřebná doba plnění je vypočítána následujícím způsobem:

```
uint8_t width = WidthToCounterVal[ ((uint16_t)amplitude* (uint16_t)SinusAprox[TCNT0]) >> 8];
```

Výpis 1: Výpočet doby plnění

Kde:

- *TCNT0* drží informaci o fázi.
- *SinusAprox* Vrací hodnotu funkce sinus pro daný parametr *TCNT0*.
- *amplitude* je požadovaná amplituda daná zpětnou vazbou.
- Součin 8 bitového sinu a 8 bitové amplitudy má velikost 16 bitů, provedením bitového posuvu doprava je hodnota upravena opět na maximální rozsah 8 bitů, jde o provedení normalizace.
- pole *WidthToCounterVal* slouží k přepočtu 8 bitové hodnoty požadované šířky plnění na potřebnou hodnotu čítače. Toto pole používáno zvláště při generování 50 Hz PWM. Pokud pracuje čítač jako 8 bitový, není potřebné.

Všechny tyto části a jejich přesné použití je popsáno dále v textu.

6.4.1 Aproximace funkce sinus

Aby bylo možné se vyhnout náročnému výpočtu sinu proběhne před spuštěním generování předvyplnění aproximační tabulky. Vstup může nabývat hodnot 0 - 255, výstupní hodnota je požadována ve stejném rozsahu. Místo náročného výpočtu stačí přistoupit do pole *SinusAprox* s indexem odpovídajícím vstupu. V poli jsou uloženy pouze hodnoty z rozsahu 0 - π , pro účely programu není nutné více. Aproximace by mohla být uložena v programové paměti nebo v EEPROM. Výhodou uložení v paměti je možnost velmi snadno dodělat podporu více druhů generovaných průběhů. Firmware je možno upravit například tak, aby aproximaci průběhu bylo možné předat pomocí RS-232 a tím uživatelsky volit tvar průběhu.

```
void initSinusAprox() {
    for(uint16_t i=0;i<255;i++) {
        SinusAprox[i] = (unsigned char) 255 * sin( i*(M_PI / 256));
    }
}
```

Výpis 2: Naplnění pole s aproximací funkce sinus

6.4.2 Externí generátor hodin

Pro vygenerování průběhu je nutné znát okamžitou polohu a fázi průběhu ve které se nacházíme. Tuto informaci udržuje čítač 0. Kdy hodnota 0 odpovídá úhlu 0 a hodnota 255 odpovídá hodnotě π . Tento časovač je řízen externím generátorem hodin s frekvencí:

$$f_{ext} = 50Hz * 2 * 256 = 25,6kHz$$

Výhodou je možnost velmi pružně měnit frekvenci modulovaného signálu, výstupní frekvence modulovaného sinu je závislá pouze na externím generátoru hodin, také je zachováno rozlišení 8 bitů a tím i zjednodušena výpočetní složitost. Při změně frekvence modulovaného signálu zůstává modulační frekvence stejná.

Čítač se používá jen pro generování modulovaného sinu pomocí PWM. Pro druhý mód chodu není potřeba. Je nastaven na externí zdroj hodin, který je přiveden na pin *T0*. Přičtení je provedeno se vzestupnou hranou hodinového signálu.

V konstrukci plní funkci externího generátoru hodin malý MCU Atmel Mega 8, který lze využít k doplnění případných funkcí, například k synchronizaci s externím zdrojem modulačního kmitočtu - veřejnou rozvodnou sítí. MCU by bylo možné nahradit například astabilním multivibrátorem s požadovanou frekvencí.

Nastavení a spuštění čítače *T0* je prováděno ve funkci *initExternal50HzCounter()*, zastavení čítače je možné pomocí funkce *stopExternal50HzCounter*.

6.5 Modulace 50 Hz PWM s mezikrokem

Firmware je schopen generovat i průběh, který používají běžné UPS. Generování je opět plně hardwarové a využívá opět čítač *T1*, ovšem jeho frekvence je snížena děličem kmitočtu 64 oproti hodinovému kmitočtu. Čítač pracuje na frekvenci:

$$f_{T1} = \frac{f_{osc}}{prescaler} = \frac{16000000}{64} = 250kHz$$

Při zachování minimálně 8 bitového rozlišení lze generovat frekvence v rozsahu:

$$f_g < \frac{f_{T1}}{2 * (2^8 - 1)} < 490,19Hz$$

$$f_g > \frac{f_{T1}}{2 * (2^{16} - 1)} > 1,90Hz$$

Firmware počítá s možností změny generované frekvence pomocí funkce *changeSteppedFreq(uint16_t freq)* v rozsahu 10 - 150 Hz, kde parametr *freq* je očekáván jako desetinásobek požadované frekvence. Software opět nemusí počítat s desetinnými čísly a je možno nastavit generovanou frekvenci s přesností na $\frac{1}{10}$ Hz.

Časovač používá mód fázově správné PWM kde maximální hodnota čítání je uložena v registru *ICR1*, hodnota tohoto registru přímo ovlivňuje generovanou frekvenci, na rozdíl od modulovaného sinu, kde je použit časovač *T0* s externím časováním. Díky tomuto módu počítá čítač do maxima opět dvakrát, proto je ve výše uvedených vzorcích $2 * \text{maximální hodnota}$. Potřebná doba plnění je stejně jako u generování PWM sinu zapsána do registrů *OCR1A* a *OCR1B*.

6.5.1 Přepočet šířky plnění na rozsah čítače

Je zřejmé že v tomto případě může být doba plnění z rozsahu 255 - 65535, celá vnitřní logika pracuje s 8 bitovými čísly, aby bylo možné se vyhnout potřebnému přepočtu je v

paměti vypočítáno pole *WidthToCounterVal*. Pole je naplněno metodou *initWidthToCounterVal*, která očekává hodnotu, do které čítač počítá, tedy hodnotu ICR1 registru. Hodnota je také dostupná v globální proměnné *stepped_lenght*. Metoda pro naplnění pole bere v potaz, že není z bezpečnostních důvodů možné použít dobu plnění 100%. Maximální šířku plnění, které bude PWM dosahovat lze nastavit v makru *SAFE_WIDTH*, výchozí hodnotou je 0,98 tedy 98%.

Pole je používáno i při generování modulovaného sinu, i když by to z principu funkce nebylo potřebné. S malou úpravou by bylo možné upravit firmware tak, aby bylo možné měnit frekvenci PWM. Není však opodstatněný požadavek k tomu, aby frekvence byla snížena pod 31,3 kHz nebo dokonce pod hranici akustického pásma 20 kHz.

6.6 Generování polovin vln

Jak jde vidět v předchozí části OCR1A a OCR1B jsou plněny stejnými hodnotami, oba dva výstupy by produkovaly naprosto shodné průběhy, což by vedlo k současnému sepnutí protilehlých prvků a následnému magnetickému zkratu. Je nutné tedy správně přepínat výstupy mezi jednotlivými generovanými polovinami.

Na konci generování jedné půl vlny je nutno zablokovat jeden z výstupu a odblokovat výstup druhý. Tento okamžik je při modulování sinu pomocí PWM signalizován přerušením při přetečení časovače *T0*, který drží informaci o momentální fázi sinu. Při generování 50 Hz PWM je konec jedné půl vlny signalizován přerušením při přetečení samotného čítače *T1*. Obě přerušování ale vykonávají ten stejný algoritmus, proto je lze obsluhovat pomocí jednoho vektoru. U MCU AVR k tomu slouží možnost definovat obsluhu přerušování jako alias jiného přerušování pomocí makra *ISR_ALIASOF*.

Samotnou záměnu výstupů obstarává funkce *swapOutputs*, informace o tom, zda je momentálně generovaná kladná nebo záporná polovina vlny udržuje globální proměnná *half_wave*.

6.7 Přechod mezi modulacemi

Součástí firmware je možnost přechodu mezi oběma způsoby výstupní modulace. Postup při změně je následující:

1. Odpojit vstupní relé
2. Odstranit staré nastavení čítačů používaných pro generování
3. Nastavit nový mód
4. Přepočítat potřebná pole v paměti (*WidthToCounterVal*)
5. Nastavit obslužné registry čítačů do potřebných hodnot
6. Připojit vstupní relé
7. Spustit čítače

6.8 Měření proudu a napětí

Snímání provozních veličin je nutné minimálně z důvody potřeby zpětné vazby. Je taky vhodné, aby uživateli byly dostupné informace o zatížení, proudu a napětí baterie. Pokud napětí olověné baterie poklesne pod 10 V dochází k poškození baterie a zkrácení životnosti.

6.8.1 Měřené veličiny a požadovaná přesnost

Měřená veličina	Požadovaný rozsah	Přesnost	Galvanické oddělení
Výstupní napětí	0 - 250 V _{ef}	10 bit	Ano
Výstupní proud	0 - 160 mA	8 bit	Ano
Napětí baterie	0 - 14 V _{DC}	8 bit	Ne

Tabulka 3: Měřené veličiny a jejich rozsah.

6.8.2 ADC převodník

Pro převod analogových signálů na digitální je Mega 644PA vybavena aproximačním převodníkem schopným pracovat s hodinovým signálem až 200 kHz při plném rozlišení 10 bitů. Převodníku je předřazen 8 kanálový multiplexer, lze tedy přepínat mezi 8 měřenými vstupy. Volba vstupu je možná pomocí bitů *MUX* v registru *ADMUX*.

Měření je možné v rozsahu 0 V a referenčním napětím do maximální velikosti napájecího. Je možno použít vnitřní zdroj referenčního napětí 1,1 V, 2,56 V nebo je možno připojit externí referenci na vývod *AREF*.

Hodinový signál převodníku je závislý na taktování mikrokontroléru. Pro přizpůsobení hodin je k dispozici 7 bitový dělič kmitočtu, volba dělicího poměru se provádí pomocí bitů *ADPS* v registru *ADCSRA*. Pro pracovní frekvenci 16 MHz je nutno volit dělicí poměr 128, výsledná pracovní frekvence převodníku je 125 kHz. Dokončení převodu lze obsluhovat pomocí přerušení s vektorem *ADC_vect*.

Spouštění měření je možné několika způsoby:

- **Free running mode** - po dokončení převodu je automaticky spuštěno další měření
- **Single mode** - po zapsání log. 1 na bit *ADSC* registru *ADCSRA* je spuštěn jeden převod.
- **Triggered mode** - měření je možné spustit pomocí přerušení od analogového komparátoru, externího přerušení nebo přerušení čítačů.

Pro měření je použit *Single mode*. Jsou měřeny celkem postupně tři analogové kanály. Měření je obsluhováno přerušením. Při příchodu přerušení od ADC jsou zpracovány naměřené informace a je přepnut multiplexer na další kanál, následuje spuštění dalšího měření.

6.8.3 Měření napětí

ADC převodník je vhodný k přímému měření napětí. Musí být dodrženo napětí v rozsahu $0 - V_{CC}$. Nejjednodušším druhem přizpůsobení je vysoko-impedanční dělič napětí, který je vhodný pro nízká stejnosměrná napětí známé velikosti. Pro napětí větší než nízká je nutno použít oddělení optočlenem, nebo v případě střídavého napětí transformátor.

6.8.3.1 Baterie Napětí baterie je odebíráno před ochranným relé a přes dělič napětí přivedeno na vstup převodníku. Dělič je navržen tak aby při napětí baterie 14 V bylo výstupní napětí děliče 5 V. Měřená hodnota je ukládána do pole o 256 hodnotách pro lepšího stanovení hodnoty, z důvodu kolísání napětí baterie při zatížení. Původní varianta používala průměrnou hodnotu. Při měření bylo zjištěno, že použití maximální hodnoty z pole dává lepší výsledky. Ve firmwaru jsou zachovány obě dvě funkce *GetBatteryAvgVoltage* i *GetBatteryMaxVoltage*.

Pro tyto účely není nutné velmi přesné měření, postačuje rozlišení 8 bitů. Z původně 10 bitové hodnoty jsou zanedbány dva bity s nejnižší významností.

Přepočet na volty probíhá pomocí vzorce 4. Výsledná hodnota je desetinásobkem hodnoty ve voltech. Výhodou je přesnost na jedno desetinné místo s použitím celočíselného datového typu.

$$\begin{aligned} 10U &= \frac{AVG(measured_values) * 255}{140} \\ 10U &= \frac{MAX(measured_values) * 255}{140} \end{aligned} \quad (4)$$

Výsledky měření měniče byly porovnávány s měřícím přístrojem DT9205A, při nezatíženém měniči naměřená hodnota odpovídá hodnotě zobrazované měřícím přístrojem, při větším zatížení měniče a tím i baterie a poklesu napětí k 10 V stoupá odchylka měření měniče a v nejhorším naměřeném případě byla odchylka -0,8 V. Lze to přikládat pulznímu zatížení, kdy napětí baterie může velmi rychle kolísat.

6.8.3.2 Výstup Měnič je konstruován pro dva možné režimy, při použití pouze jediného výstupního režimu by pro měření efektivního napětí bylo možno použít operační zesilovač, při změně tvaru výstupního napětí ale přesnost takového měření velmi klesá.

Při přímém měření pomocí AD převodníku je nutno ze vzorků skutečnou střední hodnotu vypočítat. Výpočet skutečné střední hodnoty z n vzorků lze provést podle následující rovnice:

$$x_{rms} = \sqrt{\frac{1}{n} (x_1^2 + x_2^2 + \dots + x_n^2)}$$

Výstup z transformátoru je přiveden na vstup DA převodníku, kde pomocí dvou stejných odporů je vytvořen střed, vůči kterému je napětí měřené vztaženo. Hodnota offsetu je v software definován makrem *VOLTAGE_ADC_BIAS* vhodné nastavení offsetu je 511, aby byla dodržena co největší přesnost směrem do kladných i záporných čísel. Od naměřené hodnoty je odečten offset a naměřené hodnoty se tedy ve výchozím nastavení pohybují v rozsahu -511 až +511.

Počet vzorků, ze kterých výpočet probíhá je definován v makru *VOLTAGE_SAMPLES*, vzorky jsou 16 bitové a vzrůstající množství vzorků má vliv na velikost využití paměti a dobu výpočtu efektivní hodnoty. Výchozí hodnota je 1024 vzorků. Výpočet efektivní hodnoty provádí funkce *getOutputRMS()*.

6.8.4 Měření proudu

Proud je snímán na měřicím odporu velikosti 60 Ω, pomocí kterého je proud procházející obvodem převeden na napětí, které je pak usměrněno, vyfiltrováno a na úroveň 5 V převedeno pomocí optického členu SFH 615. Optočlen pracuje v invertujícím režimu. Při nulovém proudu na primární straně je výstupní tranzistor plně uzavřen a na analogovém vstupu mikrokontroléru je napětí 5 V, které odpovídá hodnotě 255.

Samotný převod napětí pomocí optočlenu je nelineární. V první fázi bylo třeba provést měření závislosti mezi proudem procházejícím obvodem a hodnotou převodu ADC. Graf měření je uveden v příloze C. Linearizace je provedena proložením celkem třemi přímkami. Grafické znázornění proložení je obsaženo v příloze C.

$$f(x) = \begin{cases} 255 & x < 68 \\ -3,3750x + 396 & x \in (68, 92) \\ -1,0714x + 184 & x \in (92, 120) \\ -0,4110x + 108 & x \in (120, 255) \\ 0 & x = 255 \end{cases} \quad x, y \in N + 0 \quad (5)$$

Při zapnutí funkce *initADCTomA* provede pomocí rovnic 5 naplnění tabulky *ADCTomA*. Hodnota 255 je volena jako přetížení, pokud je tedy naměřená hodnota mimo horní rozsah bude přetčeno z pole toto číslo. Měření proudu je navrženo pro rozsah 0 - 167 mA.

6.9 Regulace výstupu

Software snímá výstupní hodnoty, vyhodnocuje a na základě požadované referenční hodnoty určuje potřebnou amplitudu průběhu při generování modulovaného sinu a potřebnou dobu plnění při generování 50 Hz PWM. Mikrokontrolér tak uzavírá regulační smyčku.

Referenční hodnotu může nastavit přímo uživatel pomocí displaye nebo pomocí rozhraní RS-232.

Pro výpočet velikosti amplitudy je použit rychlostní PI algoritmus, který se od rychlostního PID algoritmu, popsaného v 4.4.3, liší absencí derivační složky. Výpočet nastavovací hodnoty probíhá v okamžiku dokončení generování půl vln v přerušení *TIMER0_OVF_vect*, postup výpočtu je uveden ve výpisu programu 3.

```
Measure_now_RMS = getOutputRMS();
int16_t error = Measure_reference - Measure_now_RMS;

int16_t delta = Kp * (error - Measure_last_error) + (Ki * error);
```

Výpis 3: Výpočet změny nastavovací hodnoty

Jednotlivé parametry PI regulátoru lze uživatelsky nastavit. Výchozí nastavení bylo provedeno experimentálně pomocí Ziegler–Nicholsovy metody.

7 Návrh komunikace s počítačem, konfigurace a měření sledovaných parametrů

Mikrokontrolér Atmel Mega 644PA disponuje 2 univerzálními sériovými porty kompatibilních, které jsou až na logické úrovni plně kompatibilní se standardem RS-232.

Ačkoli byl standard RS-232 v posledních letech z domény počítačů pro běžné uživatele vytlačen rozhraním USB, tak je široce podporován ve velkém množství zařízení, téměř každý osobní počítač je jím stále vybaven a pokud není port přímo vyveden na zadní stěnu počítače, tak ve většině případů je na základní desce přítomen 10 pinový konektor ke kterému stačí dokoupit záslepku s výstupním portem. V průmyslu jsou stále velmi oblíbené rozhraní RS-422 a RS-485, které jsou velmi blízké RS-232, jsou odolné proti rušení a lze je snadno přenášet i na větší vzdálenosti.

Všechny tyto faktory jsou rozhodující pro použití RS-232 pro komunikaci s aplikací postavenou na bázi mikrokontroléru.

7.1 RS-232

První specifikace vznikla v roce 1962 pro účely dálkopisu. V dnešní době je používána revize F z roku 1997. Ve specifikaci jsou definovány části jako elektrické vlastnosti, linkové úrovně přenosové rychlosti a časování sběrnice, mechanické vlastnosti, konektory ale nedefinuje části jako kódování znaků nebo způsoby detekce chyb.

7.1.1 Omezení standardu

Díky jisté „zastaralosti“ a tomu, že dnešní použití dalece překročilo původně zamýšlený rozsah, existují jistá omezení, kvůli kterým bylo ve spotřební elektronice vytlačeno rozhraním USB. Například:

- Relativně velká napětí jednotlivých logických úrovní zvyšují nároky na napájení a také omezují maximální možnou rychlost.
- Vztažení úrovní vůči společné zemi snižuje odolnost proti rušení.
- Bez možnosti napájení zařízení na druhé straně.
- Nízká maximální přenosová rychlost

7.1.2 Napěťové úrovně

Základní tři vodiče, které jsou pro duplexní komunikaci potřebné jsou *RxD* a *TxD* a země (*GND*). Pro datové linky *RxD* a *TxD* je definována logická 0 jako napětí v rozsahu +3V až +15V a logická 1 jako napětí -3V až -15V. Dále jsou na rozhraní přítomny řídící linky *RTS*, *CTS*, *DTR*, *DSR*, ... Rozhraní je zkratu vzdorné, maximální proud je omezen na 20mA.

Modulační rychlost (Bd)	Maximální délka (m)
19 200	15
9 600	150
4 800	300
2 400	900

Tabulka 4: Závislost délky vedení na modulační rychlosti [6].

Synchronní

Vhodné pro velké objemy dat

Nutno určit, kdo bude vysílat synchronizační impulsy

Jednoduchá elektronika

Vodič navíc, který nepřenáší data

Asynchronní

Nevhodné pro velké objemy ale vhodné pro velké vzdálenosti

Nutno přesně definovat přenosovou rychlost

Složitá elektronika s krystalovým oscilátorem

Menší přenosová rychlost kvůli synchronizačním a paritním bitům

Tabulka 5: Porovnání synchronního a asynchronního přenosu u RS-232

7.1.3 Mechanická specifikace

Zařízení připojena na rozhraní, můžeme rozdělit na zařízení ukončující datový okruh (DCE) nebo koncové zařízení přenosu dat (DTE). Podle typu zařízení jsou specifikovány typy konektorů a rozložení jednotlivých pinů. Standard doporučuje použití 25 pinového konektoru D-sub, kde na zařízení DTE je osazen samec a na zařízení DCE je osazena samice. Na osobních počítačích bývá většinou osazen konektor samec DE-9. U některých výrobců může být použit i konektor 8P8C, ten bývá osazen například u směrovačů firmy Cisco.

Specifikace neurčuje maximální délku kabeláže, ale určuje maximální kapacitu vedení 2500 pF. Všeobecně je doporučováno používat vedení o maximální délce 15m. Doporučované délky ve vztahu s modulační rychlostí jsou uvedeny v tabulce 4.

7.1.4 Časování sběrnice

RS-232 podporuje jak synchronní přenos tak asynchronní. Při synchronní komunikaci je nutno použít vodič navíc, který přenáší synchronizační impulzy. Při asynchronním přenosu je nutno znát předem přesnou přenosovou rychlost. Data začínají startovací sekvencí, vůči které jsou data potom synchronizována. Rychlé srovnání synchronního a asynchronního přenosu je uvedeno v tabulce 5.

Lze přenášet 5, 7, 8, 9 bitová data od nejméně významného po nejvíce významný. U asynchronního přenosu začíná přenos jedním start bitem, po přenosu dat může následovat paritní bit a potom vždy jeden nebo více stop bitů. U synchronního přenosu je pomocí synchronizačního vodiče signalizován přijímači přenos dat.

V běžném PC je maximální rychlost sériového portu 115200 Bd a ostatní rychlosti jsou odvozeny dělením této hodnoty. Běžně jsou používány tyto modulační rychlosti: 115200 Bd, 57600 Bd, 38400 Bd, 19200 Bd, 9600 Bd, 4800 Bd, 2400 Bd. Je zřejmé, že z důvodu přítomnosti start, stop a paritních bitů bude přenosová rychlost vždy menší.

7.1.4.1 Parita je jedním z nejjednodušších způsobů jak bez velkých nároků na výpočetní výkon zabezpečit přenos dat. Vysílací zařízení sečte počet jedničkových bitů a doplní se paritním bitem tak, aby byla zachována předem dohodnutá podmínka sudého nebo lichého počtu jedničkových bitů.

7.2 Návrh komunikačního protokolu

Již víme, že data budeme přenášet pomocí protokolu RS-232, je však nutno nadefinovat, jak budou data interpretována. Jednotlivé strany se musí dorozumívat na základě pevně daného protokolu. Obecně můžeme stanovit dva způsoby komunikace.

7.2.1 Bitově orientovaný protokol

Při bitově orientovaném protokolu je definice významu přeneseného bitu nebo bytu čistě na vůli výrobce. Může být například stanoveno, že délka zprávy je 16 bitů, kde prvních šest bitů je kód operace a zbývajících 10 je hodnota parametru operace. Je zřejmé že u takového kódování může parametr nabývat 1024 celkem hodnot, jak tento rozsah využijeme, je jen na nás. Zpráva je často logicky rozdělena na hlavičku - často s pevnou délkou, data a patičku, která obsahuje například kontrolní součty. Výhodou binární formátu je možnost dosáhnout vysoké kompaktnosti přenášených dat a tím i velkých přenosových rychlostí.

7.2.2 Znakově orientovaný protokol

Zpráva se skládá ze znaků, kde každý z nich je v daném kódování. Pro kódování může být použita například ASCII tabulka, která potřebuje pro zakódování jednoho symbolu 7 nebo 8 bitů. Budeme-li chtít přenést stejně jako u předchozího příkladu číslo 1024, bude potřeba celkem 4 bytů. Výhodou je snadná čitelnost přenášených dat, protože kódování jako ASCII nebo UTF-8 jsou všeobecně známá. Snadno se také hledají problémy při vývoji, protože software pro zachytávání provozu je většinou automaticky dekodují.

7.2.2.1 ASCII tabulka Je standardizovaná kódovací tabulka vyvinutá společností Bell, publikovaná ASA v roce 1967. V sedmi bitech kóduje 128 znaků. Je možné zakódovat číslice 0 až 9, velké a malé anglické abecedy, interpunkční znaménka, mezera a několik z kontrolních kódů pro řízení dálnopisných zařízení, některé z nich jsou velmi dobře známy i v jazyce C jako například `\n` pro nový řádek nebo `\t` pro tabulátor. V dnešní době se používá 8 bitů, dalších 127 znaků se používá pro přenos národních abeced. I přesto je 256 symbolů pro zakódování všech národních symbolů nestačí a pro tyto potřeby je vytvořen organizací ISO systém kódovacích tabulek. Význam dalších 127 je tedy nejednoznačný.

7.2.3 Návrh protokolu

Pro jednoduchost a snadné řešení problémů při vývoji je komunikační protokol s měničem navržen jako znakově orientovaný s kódováním pomocí ASCII tabulky. Iniciátorem komunikace je vždy počítač a komunikace bude probíhat formou otázka odpověď s tím, že pokud je zpráva správně přijata, tak firmware měniče vždy odpoví.

Časování sběrnice je asynchronní s modulační rychlostí 4800 Bd. Datový rámec je 8 bitový bez paritních bitů.

7.2.3.1 Formát zpráv počítače Firmware očekává zprávu uvedenou hodnotou *Start Of Text*, tedy v hexadecimální podobě 0x02. Zprávy jsou ukončeny znakem *End of Transmission* - 0x04. Teprve po přijetí celé zprávy je předána ke zpracování. Během zpracování je příjem další zprávy zakázán. Obsah zprávy musí obsahovat na prvním místě volací znak požadované funkce. Pokud funkce vyžaduje další parametr, tak následuje hned po znaku funkce. Je-li hodnotou desetinné číslo, je desetinné místo odděleno tečkou. Některé funkce vyžadují parametr jako desetinásobek požadované hodnoty, není je nutno zadávat jako desetinná čísla.

0x02<Func code>(Param)0x04

0x02	Start Of Text
<Func code>	Volací znak funkce
(Param)	Parametr (nepovinný)
0x04	End of Transmission

7.2.3.2 Formát zpráv firmware Na každý požadavek, který má správný formát měnič odpoví. Pokud je položen dotaz na stavovou proměnnou, tak odpovídá přímo hodnotou požadované proměnné. Pokud se jedná o nastavovací funkci a změna byla vykonána správně, tak odpoví *OK*, při nevykonání, například z důvodu parametru mimo rozsah, tak odpovídá *KO*. V případě, že formát přijaté zprávy byl správný, ale zprávu nebylo možné dekodovat nebo byla neplatná, odpoví firmware zprávou *ERR*. Všechny zprávy které měnič vysílá jsou ve výchozím stavu ukončeny znakem `\n`. Toto chová lze ovlivnit změnou makra `USART_END_OF_MSG`.

7.2.3.3 Zprávy počítače Firmware dokáže dekodovat následující zprávy:

0x02S0x04	Odpovídá typem modulace buď „SINPWM“ nebo „PWM50HZ“
0x02B0x04	Žádost o napětí baterie, odpovídá hodnotou 10x napětí baterie
0x02I0x04	Žádost o výstupní proud, odpovídá číselnou hodnotou v mA
0x02O0x04	Žádost o výstupní napětí, odpovídá hodnotou ve voltech
0x02M0x04	Změna pracovního módu
0x02P0x04	Dotaz na hodnotu konstanty proporcionality
0x02T0x04	Dotaz na hodnotu integrační konstanty
0x02K<0.000>0x04	Nastavení konstanty proporcionality
0x02N<0.000>0x04	Nastavení integrační konstanty
0x02R0x04	Dotaz na okamžitou výstupní frekvenci, dostupné jen v 50Hz PWM módu
0x02F<000>0x04	Nastavení výstupní frekvence, dostupné jen v 50Hz PWM módu
0x02E<000>0x04	Nastavení referenční hodnoty ve voltech.
0x02L0x04	Odepni ochranné relé
0x02U0x04	Sepni ochranné relé

Tabulka 6: Zprávy, které je měnič schopen zpracovat

7.3 Sériový port Atmel Mega 644PA a jeho obsluha

Univerzální synchronní/asynchronní sériové porty mikrokontroléru jsou plně duplexní s vlastním generátorem hodin odvozeným od hodin MCU. Generátor hodin má vysoké rozlišení a je možno dosáhnout velkého spektra modulačních rychlostí. Je vybaven také kontrolou rozpadu rámce, přetečením vstupního bufferu a hardwarovou kontrolou parity.

Jsou dostupné tři vektory přerušení, při přijetí bytu, vyprázdnění výstupní fronty a dokončení vysílání.

7.3.1 Inicializace sériového portu

Pro správnou funkci asynchronního přenosu je potřeba správně nastavit hodnotu baudrate registru *UBRR*. Výpočet hodnot lze ponechat na preprocesoru jazyka C. V hlavičkovém souboru *util/setbaud.h* je dostupné makro, které výpočet provede. Stačí v záhlaví programu nadefinovat hodnoty *F_CPU* s frekvencí generátoru hodin a *BAUD* s požadovanou modulační rychlostí. Hodnoty registru *UBRR* jsou pak dostupné v makrech *UBRRH_VALUE* a *UBRRL_VALUE*. Zvolený formát datového rámce odpovídá výchozímu nastavení sériového portu v MCU Atmel Mega 644PA.

Povolení příjmu (*RXEN0*), vysílání (*TXEN0*) a výběr využívaných přerušení je volen bity v registru *UCSR0B*.

7.3.2 Obsluha přijetí a vysílání bytu

Pokud je povolen příjem, je při přijetí byte vyvoláno přerušení s vektorem *USART_RX_vect*. Ve zpracovávající proceduře je nutno zkontrolovat, zda nedošlo k rozpadu rámce, nebo přetečení bufferu při nevyzvednutí bytu. Při správném přijetí je zkontrolován přijatý znak zda-li se nejedná o řídicí symbol *Start Of Transmission*, při kterém je index bufferu zprávy přesunut na začátek, nebo *End Of Transmission*, při kterém dojde k zablokování přijímače a předání bufferu zprávy ke zpracování. Nepatří-li znak k řídicím je zapsán do bufferu a ukazatel bufferu zvýšen o 1.

Vysílací proceduře je předán výstupní buffer a povoleno vysílání a s tím i přerušení při vyprázdnění výstupního registru. Po odeslání jednoho bytu je vyvoláno přerušení a je naplněn znova výstupní registr, do té doby dokud procedura nedojde ke konci zprávy umístěné ve výstupním bufferu. Pokud je procedura na konci zprávy, tak je zablokováno vysílání a povoleno přijímání zprávy.

7.3.3 Zpracování zprávy

Přerušení obstarávají jen přijetí zprávy, její další zpracování a rozkódování probíhá v hlavní smyčce. Po přijetí zprávy je další přijímání zastaveno a nastaven příznak *EOM*. V hlavní smyčce je periodicky testován tento příznak a pokud je nastaven, tak je volána funkce *DecodeMessage*. Zjednodušený diagram znázorňující zpracování je na obrázku 15.

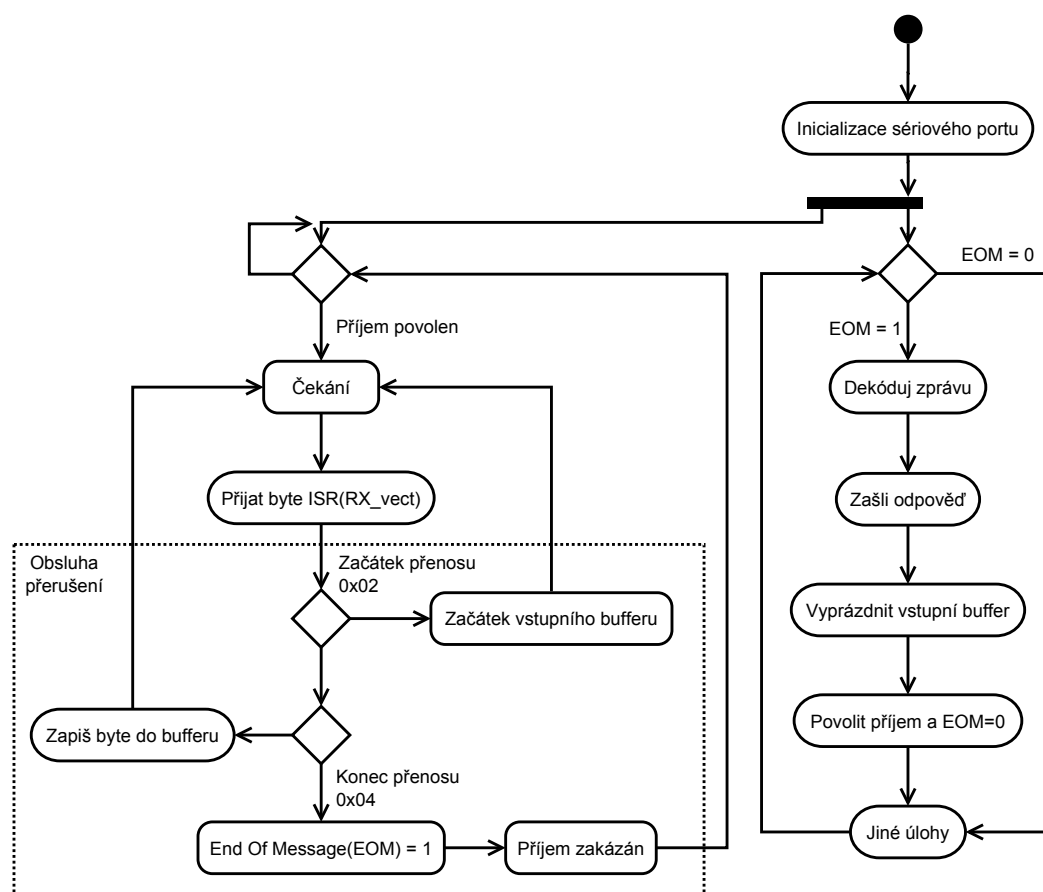
Ve funkci je podle volacího znaku vykonána požadovaná funkce. Po vykonání je sestavena zpráva odpovědi a odeslána pomocí funkce *SendMessage*. Teprve po dokončení vysílání je opětovně povolen příjem.

7.4 Display

K měniči je připojen display, na kterém jsou zobrazovány informace a je možno pomocí něj a ovládacích tlačítek nastavit provozní parametry. Display je obsluhován pomocí knihovny pro čtyř vodičovou komunikaci, která je dostupná online⁴. Firmware počítá s dvouřádkovým displayem o šestnácti znacích na řádek.

Menu je jednoúrovňové a jedno tlačítko slouží k pohybu v menu. Další dvě tlačítka jsou určena k nastavení parametrů.

⁴http://homepage.hispeed.ch/peterfleury/group__pfleury__lcd.html, dne 2.4. 2014



Obrázek 15: Obsluha přijetí zprávy sériovým portem

8 Experimentální ověření funkčnosti a stability

Ačkoli jsou v textu odděleny kapitoly o hardware a software, tak vývoj obou částí probíhal současně. Bylo nutno postupně testovat jednotlivé části HW, vždy s firmwarem upraveným pro tyto testy.

Prvně byly podrobeny testování budiče koncových spínacích prvků, budiče byly testovány do umělé zátěže proudy až 3 A při frekvenci vstupního obdélníkového signálu až 60 kHz. Cílem bylo ověřit ztrátové výkony při spínání a kvalitu vzestupné sestupné hrany. Ukázalo se, že budiče jsou schopny spolehlivě dodat potřebný proud. Vzestupná hrana obsahovala zákmit, který ale jak se ukázalo, pocházel ze zdroje obdélníkového signálu, kterým byl MCU.

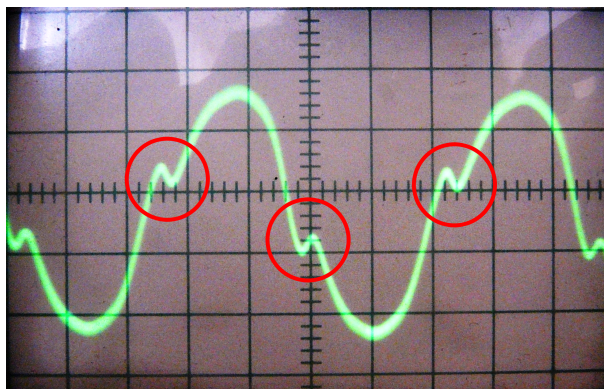
Spínací prvky byly testovány do umělé zátěže tvořené řízeným MOSFET tranzistorem. Při testování proud těmito prvky překročil 8 A při frekvenci 60 kHz. Na testovací zátěži bylo spalováno na teplo přes 100 W ztrátového výkonu. Ztrátový výkon tranzistorů byl minimální a použité chladiče byly schopny spolehlivě koncové tranzistory uchladiť.

Pro testování celého měniče byly používány jako zátěž žárovky 25 a 40 W. První z řady testů byl test generování 50 Hz PWM a generování sinusového průběhu pomocí vysokofrekvenční PWM. Během testů se ukázalo, že zvolený transformátor není pro danou aplikaci vhodný. Prvním nedostatkem bylo, že transformátor měl nedostatečné sekundární napětí, následkem je, že měnič schopen dosáhnout výstupního napětí 230 V_{ef}, ale jen asi 180 V. Dalším prvkem který má negativní vliv na chod je železný materiál jádra. Díky tomu není tento transformátor vhodný pro provoz na frekvencích v řádech kHz a dochází k saturaci jádra, který se projevuje jako zákmit na obrázku 16. Průběh generovaný samotným MCU byl také ověřen a jevil se jako naprosto bezchybný A.20. Protože transformátory pro pulzní provoz nejsou běžně k dostání a vcelku vždy se jedná o zakázkovou výrobu, která by byla velmi drahá, bylo rozhodnuto, že pro účely testování firmware je tento postačující.

Je zřejmé, že současně s těmito testy byly prováděny i testy firmware. V prvních variantách byly informace o stavu zobrazovány na displayi. Byla otestována základní schopnost generování průběhů a byl ověřen tvar generovaných průběhů. Ukázalo se, že generování průběhů musí být co nejvíce ponecháno na úrovni hardware mikrokontroléru, protože při tak vysoké frekvenci se stává režie přerušování nezanedbatelnou a může vést k viditelnému zkreslení výstupního signálu. Chod byl stabilní a bezproblémový.

Další části vývoje a testování bylo měření provozních veličin. Byly navrženy způsoby jak jednotlivé veličiny měřit, následovala kalibrace podle měřicího přístroje DT9205A, který byl nejlepší dostupný v době vývoje, případně linearizace, pokud zvolený způsob měření nezaručoval linearitu.

Pokud je firmware schopen změřit výstupní napětí, je možné vypočítat regulační odchylku a aplikovat regulační algoritmus. Samotný P algoritmus se jevil jako nedostatečný pro plynulou regulaci, protože pracuje s trvalou regulační odchylkou. PI algoritmus se projevil jako dostatečný. V tomto bodě se projevil nedostatečný výkon mikrokontroléru. MCU je schopen provádět výpočet skutečné efektivní hodnoty asi s frekvencí 100 Hz což znamená dvakrát za periodu 50 Hz. Díky tomu je regulace relativně pomalá. Pro



Obrázek 16: Zákmity způsobené transformátorem

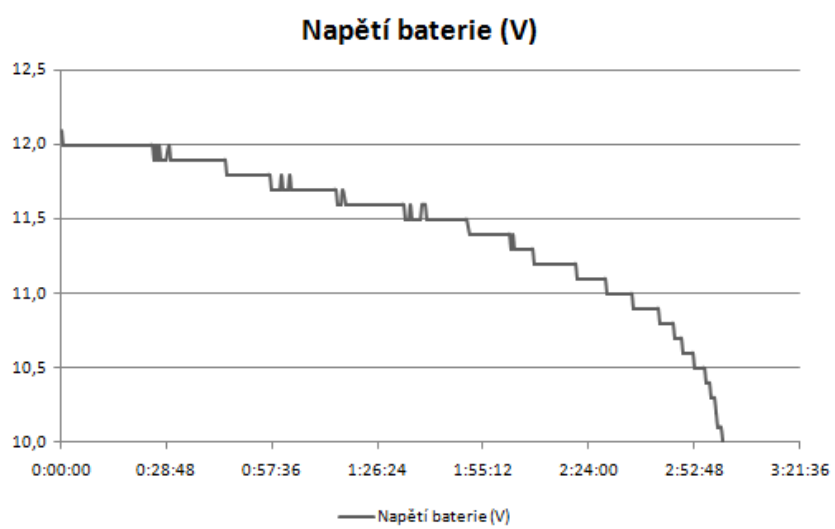
kompensaci tohoto jevu byla ponechána integrační konstanta mírně větší, což ale také způsobuje malý překmit při změně zátěže nebo požadované hodnoty.

Poslední částí bylo ověření komunikace s počítačem. Základní funkčnost komunikace byla testována pomocí převodníku USB-RS232 a programu Hercules firmy HW Group ve verzi 3.2.6. Snímek obrazovky ze zkoušení je součástí příloh B.

Po otestování jednotlivých částí bylo provedeno několik dlouhodobých testů, během kterých byla provozní data snímána pomocí rozhraní RS-232. Ukládání a pořizování dat v počítači obstarával jednoduchý program v programovacím jazyce Python využívající knihovnu PySerial. Příklad získaných provozních dat je zobrazen na obrázku 17, na kterém je znázorněno vybíjení baterie v průběhu provozu. Baterie je vybíjena na hranici 10 V, kdy zafunguje ochrana proti podvybití baterie a je odpojeno vstupní relé. Během dlouhodobých testů se provoz jevil jako stabilní a bezproblémový.

Prototyp není vhodný k plnému nasazení do provozu. Protože měnič pracuje s nevhodným transformátorem, který je navíc dimenzován na větší zatížení než je testovací, nebylo možné změřit výslednou účinnost, tyto faktory by do měření zanášely nezanedbatelnou chybu.

Pro další vývoj by bylo vhodné nahradit transformátor nebo dokonce přejít na jiný typ topologie, například zvolit dvoustupňovou konstrukci s napětovým mezistupněm, kde první stupeň zvýší napětí baterie na úroveň amplitudového napětí a druhý stupeň, H můstek, by pracoval v roli střídače. Toto řešení by ale zvýšilo požadavky na řízení, protože by bylo nutné řídit dva plnohodnotné měniče a s tím i snímat provozní stav dvou měničů. Výhodou takové konstrukce by byla ale lepší odezva na změnu zatížení. Jelikož v jednoduché topologii, která je popsána v této práci, pracuje MCU na hranici svých možností vyžádala by si změna topologie i přechod na vyšší model 16-ti nebo 32-ti bitového mikrokontroléru například Atmel XMEGA.



Obrázek 17: Graf vybíjení baterie

9 Závěr

V této diplomové práci byla popsána problematika topologií pulzních měničů vhodných pro napájení zařízení závislých na síti, která je nutným základem k pochopení principů a možností implementace řízení. Nedílnou součástí jsou také principy modulování výstupního napětí měničů a regulace pomocí pulzní šířkové modulace. V neposlední řadě bylo nutno pochopit a popsat základní prvky, postupy a algoritmy týkající se regulace a způsoby, jakými dochází ke stabilizaci výstupních veličin při náběhu nebo změnách zatížení a požadované hodnoty. Práce popsala jednoduchý způsob nastavení regulátoru pomocí heuristické Ziegler–Nicholsovy metody a následovalo ověření algoritmus v praktické části.

Pro ověření možností řízení pomocí mikrokontroléru byl sestaven konstrukčně co možná nejjednodušší měnič, na kterém byl vyvíjený firmware testován. Pro řízení byl zvolen MCU firmy Atmel spadající do kategorie středně výkonných 8 bitových mikrokontroléru. Už v prvních úvahách se tato aplikace jevila jako výkonnostně náročná, což se následně potvrdilo a následkem čeho bylo nutné firmware co nejvíce optimalizovat na výkon. Nevýhodou je, že výkon a paměťové nároky jsou spojitou nádobou a výslední firmware využívá přes 95 % z 4 kB dostupné paměti ačkoli velikost samotného programu nepřekračuje 9 kB.

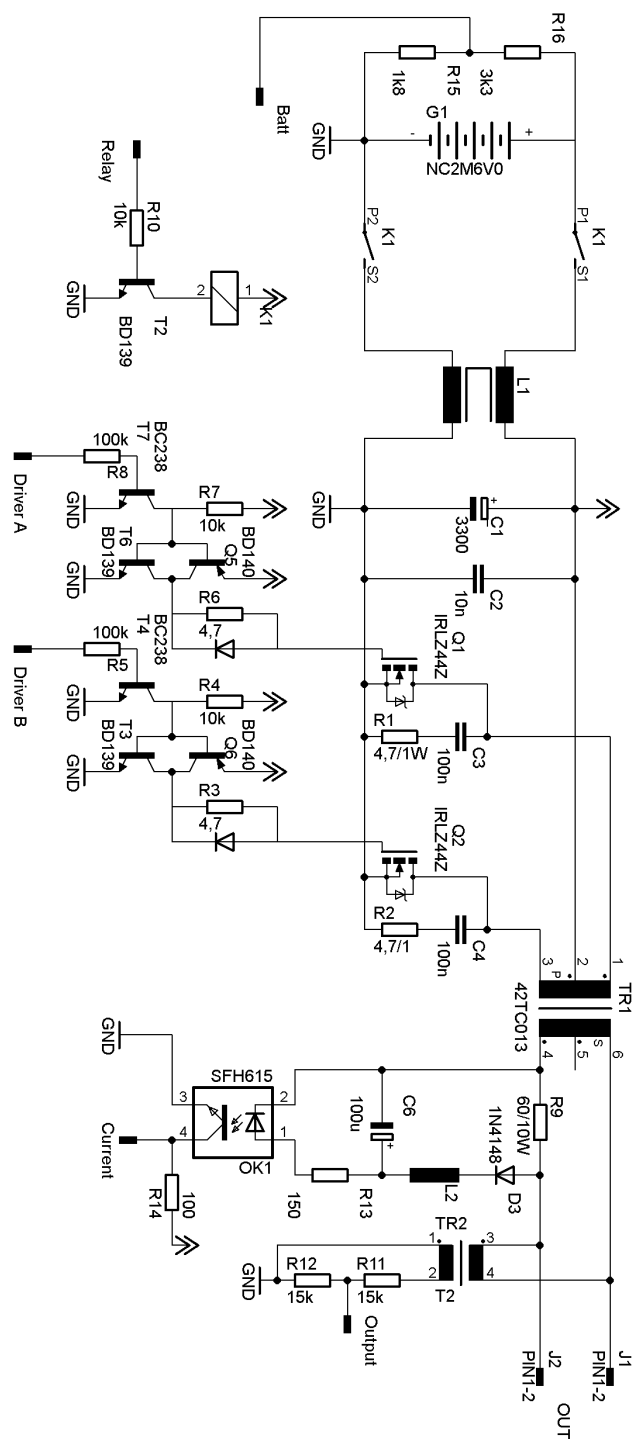
Bylo prokázáno, že 8 bitový mikrokontrolér je schopen zastat funkci řídicí prvku měniče, včetně možností regulace, snímání provozních veličin i komunikace prostřednictvím sériové linky. Přesto možnosti jsou omezené výpočetním výkonem. Při testování se ukázala nedokonalost konstrukce měniče a nepovedlo se jí odstranit. Pro další vývoj by bylo vhodné doporučit změnit hardwarovou topologii konstrukce i přechod na výkonnější MCU.

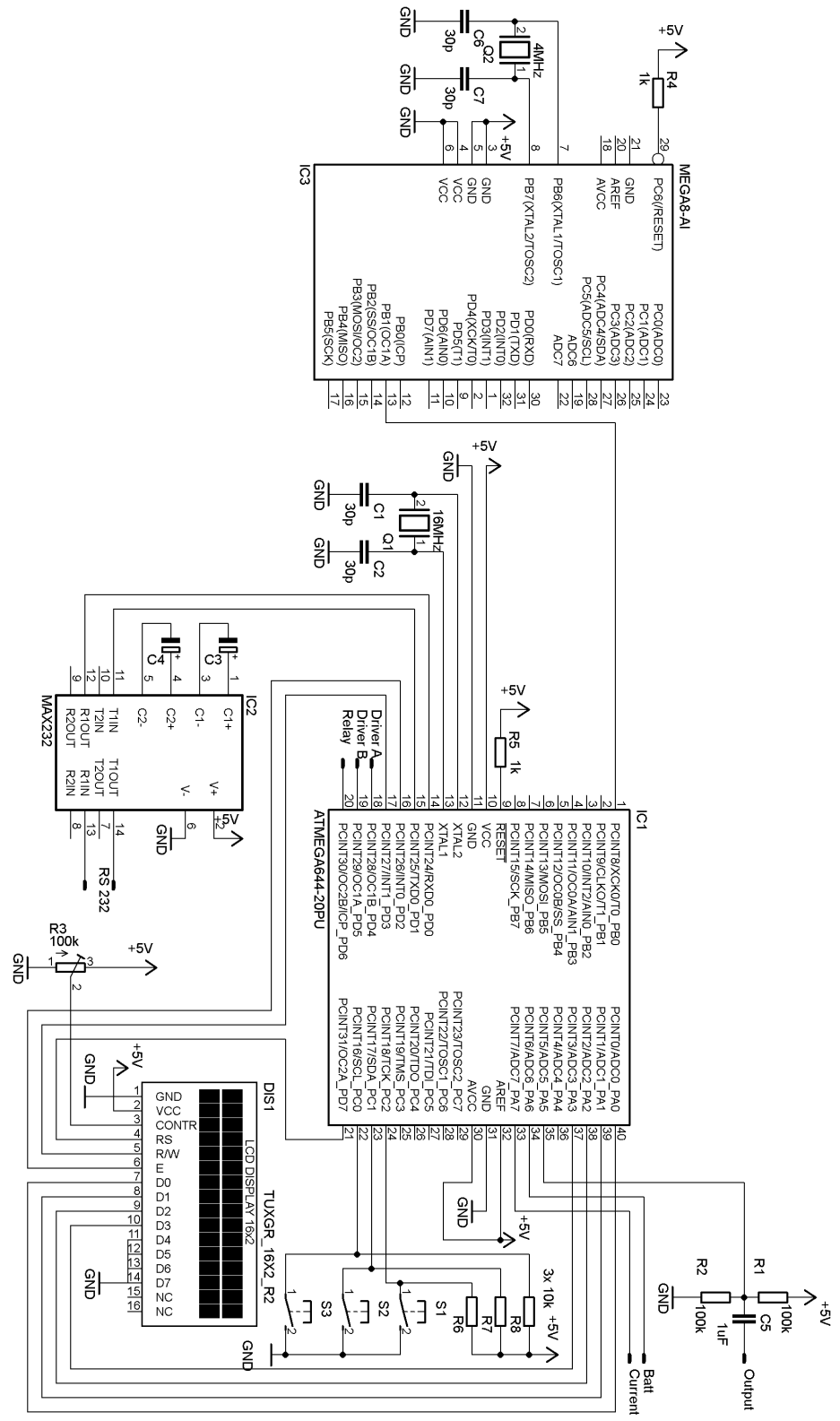
Výsledná konstrukce spolu s firmwarem napsaným v jazyce C poskytují ucelenou představu o možnostech konstrukce reálného zařízení, které by bylo možné uvést do ostrého provozu a také umožňuje případnou snadnou replikaci výsledků této práce.

10 Reference

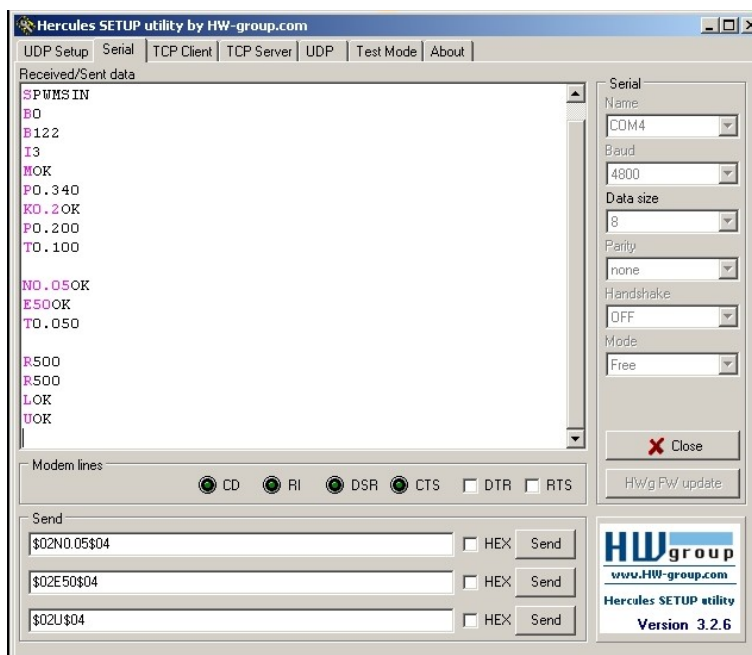
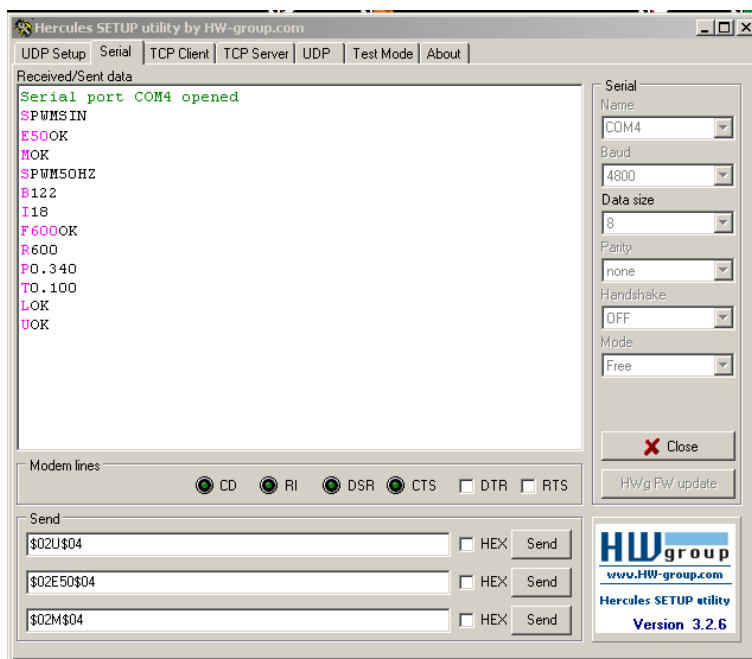
- [1] *ATmega644PA*, Atmel Corporation, 2009, URL: http://www.atmel.com/Images/Atmel-8272-8-bit-AVR-microcontroller-ATmega164A_PA-324A_PA-644A_PA-1284_P_datasheet.pdf Ver. 8152A-11/08, [citováno 10. března 2014]
- [2] *AVR200: Multiply and Divide Routines*, Atmel Corporation, 2009, URL: www.atmel.com/Images/doc0936.pdf, Ver. 0936D-AVR-09/09, [citováno 10. března 2014].
- [3] BROWN, Marty, *Practical switching power supply design*, Academic Press, 1990.
- [4] *Digital Pure Sine Wave Uninterruptible Power Supply (UPS) Reference Design*, Microchip Technology Inc., URL: <http://www.microchip.com/pagehandler/en-us/devtools/ups-reference-design.html>, Ver. B, [citováno 10. března 2014]
- [5] HÄBERLE, Heinz, *Průmyslová elektronika a informační technologie*, Europa-Sobotáles, 2003
- [6] *HW server představuje - Sériová linka RS-232*, URL: <http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>, [citováno 11. března 2014]
- [7] PRESSMAN, Abraham I. *Switching power supply design*, McGraw-Hill, 1998.
- [8] ZIEGLER, JG. - NICHOLS, NB. Optimum settings for automatic controllers. *Journal of dynamic systems measurement and control*, Červen 1993, svazek: 115, číslo: 2B strany: 220-222.

A Schéma elektrické části

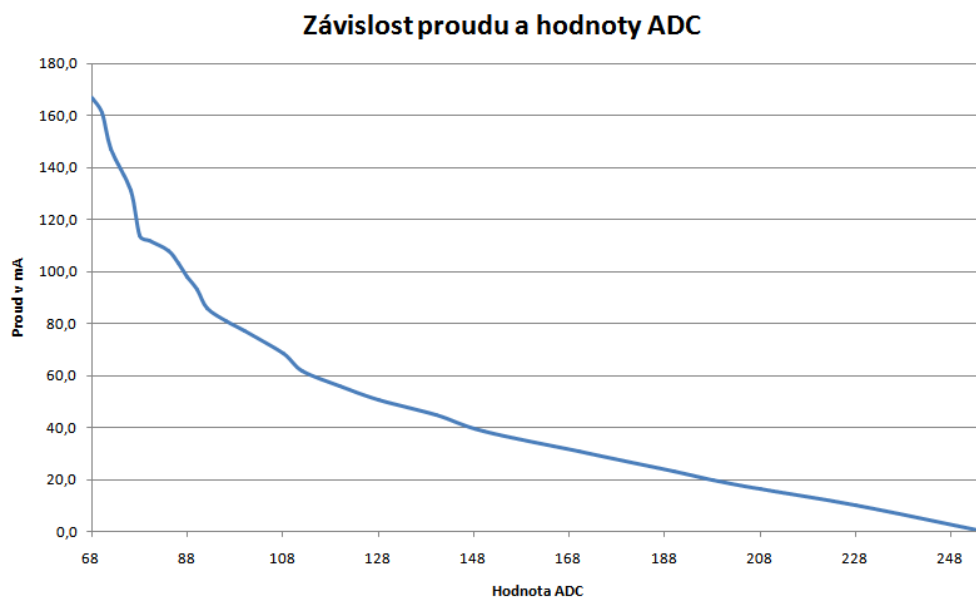




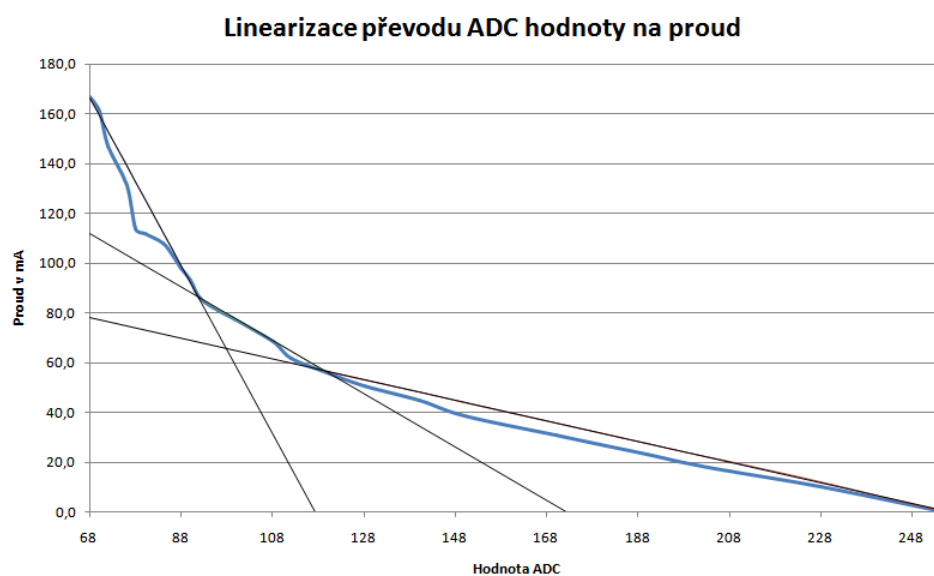
B Testování komunikace



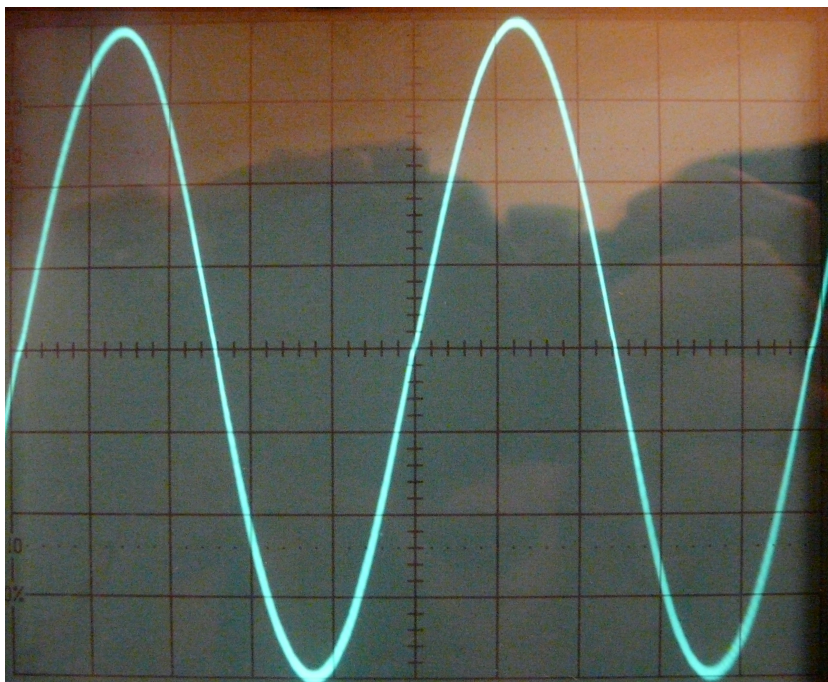
C Grafy a měření



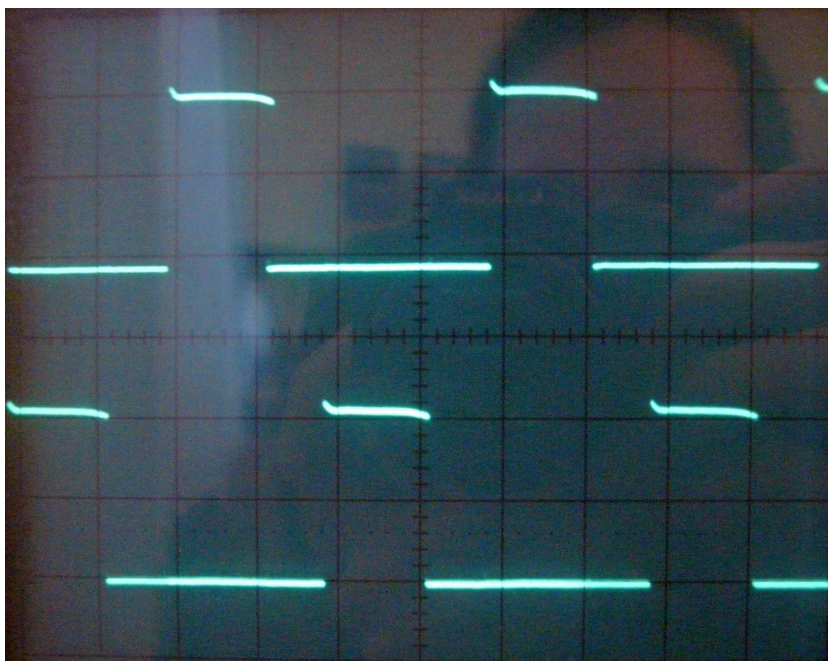
Obrázek A.18: Závislost snímané hodnoty proudu a hodnoty převodu ADC



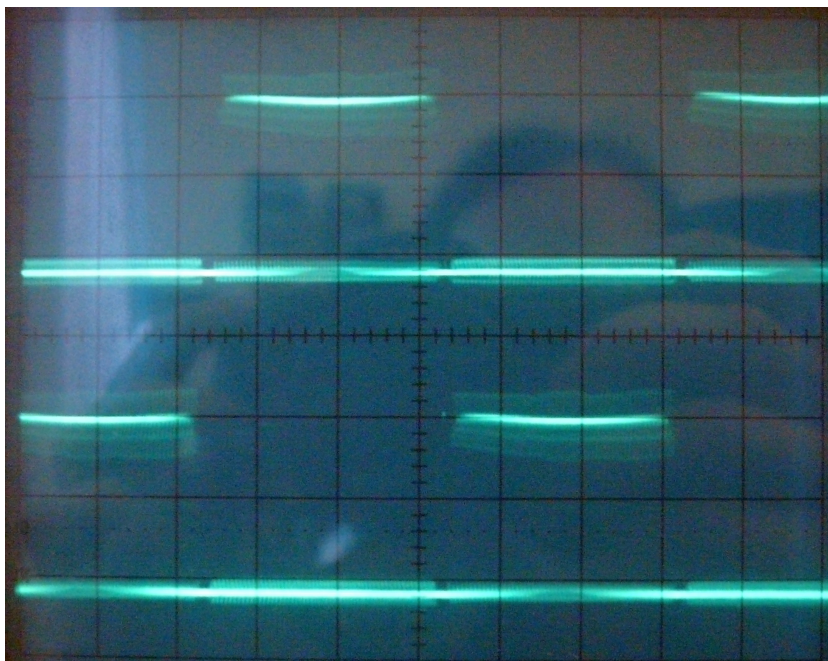
Obrázek A.19: Závislost snímané hodnoty proudu a hodnoty převodu ADC, aproximace přímkami



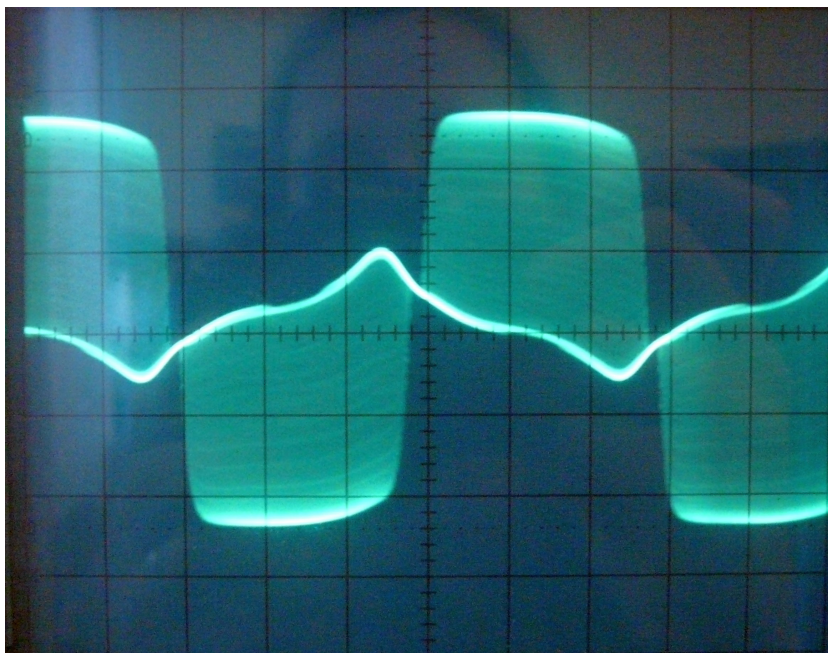
Obrázek A.20: Výstup mikrokontroléru při vysokofrekvenční PWM. Průběh je vytvořen inverzí jednoho výstupu a následným součtem.



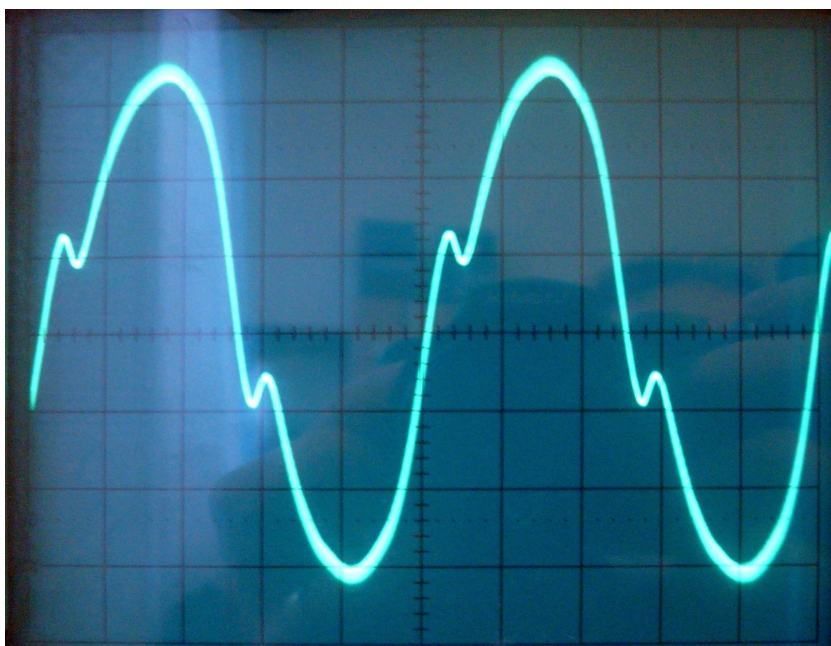
Obrázek A.21: Napětí G elektrody při 50 Hz PWM



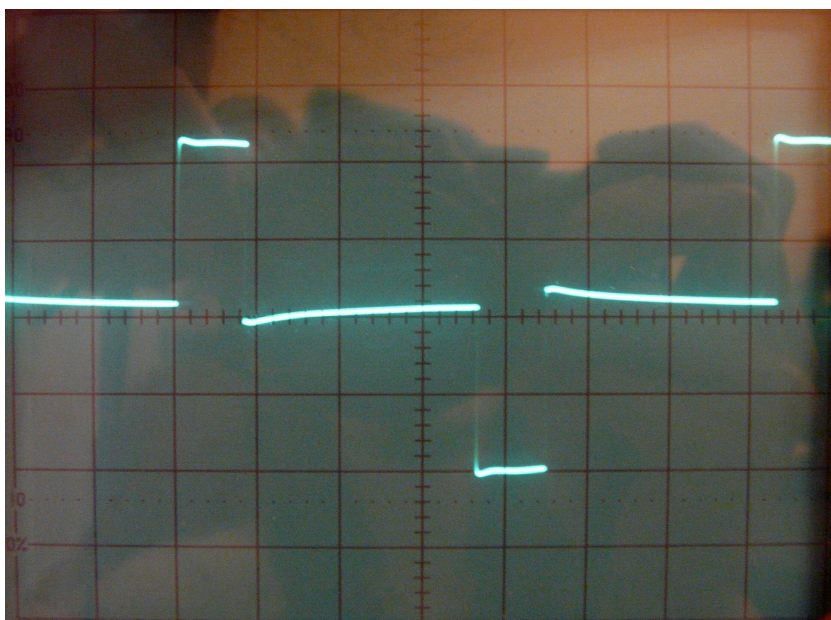
Obrázek A.22: Napětí G elektrody při modulovaném sinu



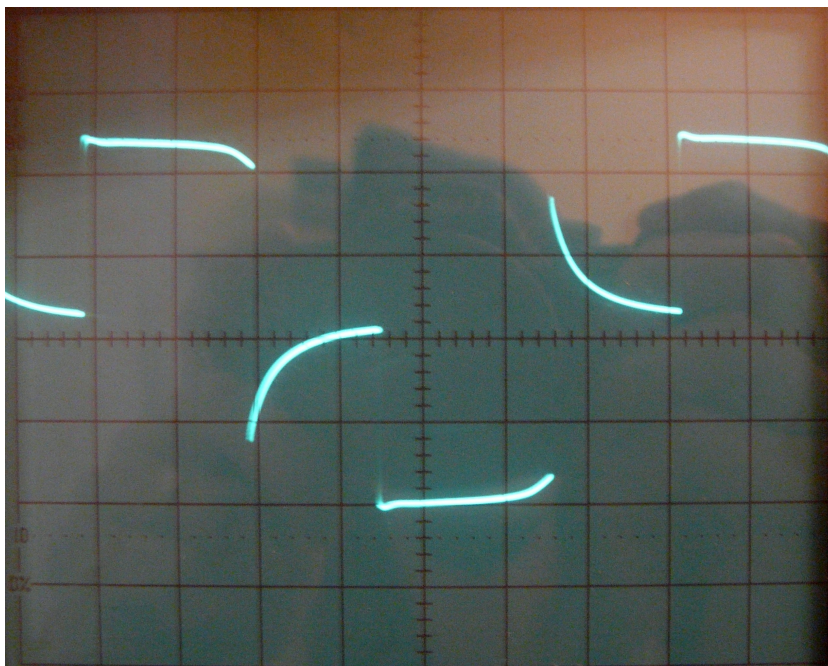
Obrázek A.23: Nevyfiltrovaný výstupní sinusový průběh



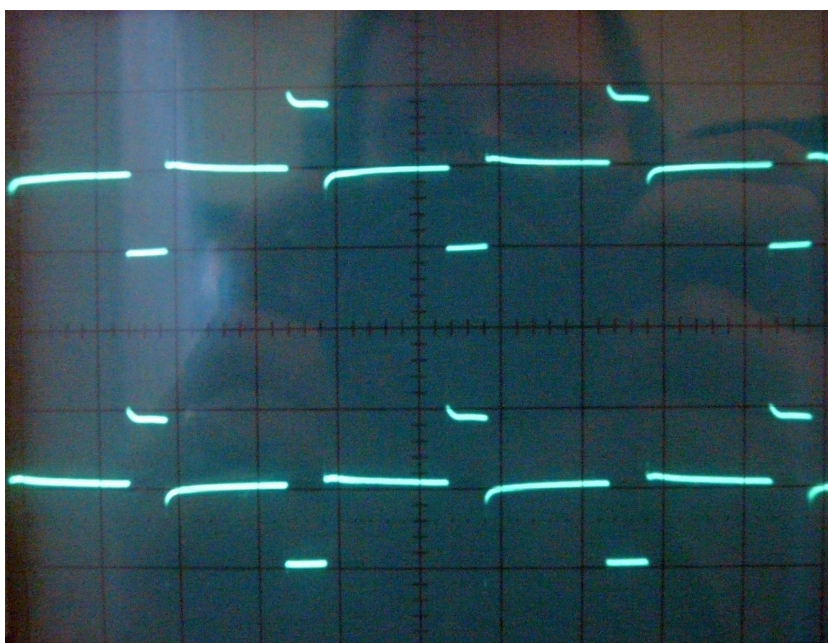
Obrázek A.24: Vyfiltrovaný výstupní sinusový průběh



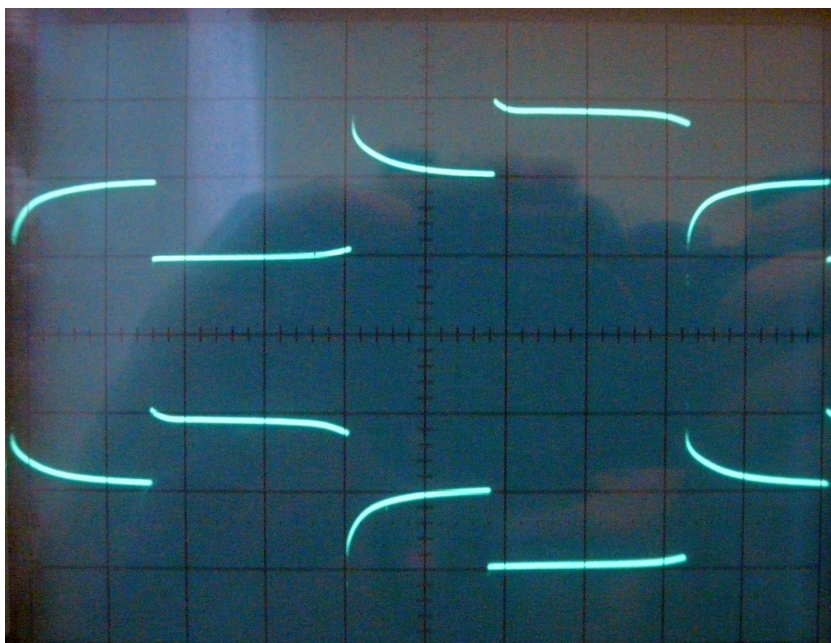
Obrázek A.25: Výstupní průběh 50Hz PWM



Obrázek A.26: Výstupní průběh 50Hz PWM, saturace jádra



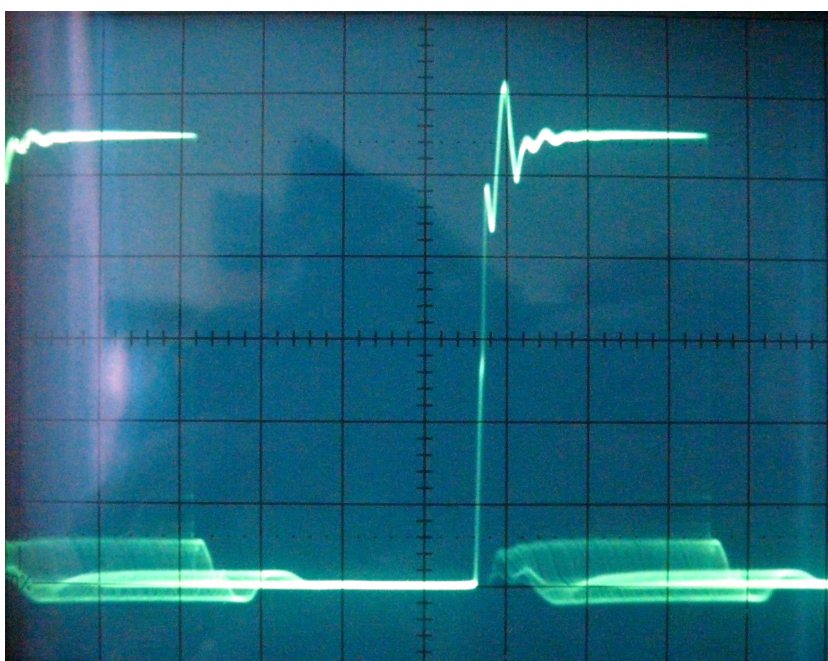
Obrázek A.27: Napětí D-S při 50Hz PWM



Obrázek A.28: Napětí D-S při 50Hz PWM, saturace jádra



Obrázek A.29: Napětí D-S při vysokofrekvenční PWM



Obrázek A.30: Výstup budiče při vysokofrekvenční PWM

D Architektura firmware

D.1 Vysoká priorita

- Generování modulovaného sinu: *TIMER0_OVF_vect*, *TIMER1_OVF_vect*
- Generování 50Hz PWM: *TIMER1_CAPT_vect*, *TIMER1_OVF_vect*

D.2 Střední priorita

- Obsluha AD převodníku: *ADC_vect*
- Obsluha sériového portu pro komunikaci s PC: *USART_UDRE_vect*, *USART_RX_vect*

D.3 Nízká priorita

- Zobrazení údajů na displayi
- Snímání tlačítek
- Změna provozních parametrů
- Řízení ochranného relé

D.4 Funkce

- *void initWidthToCounterVal(int count_to)* - inicializace pole pro přepočítání šířky
- *void initSinusAprox()* - inicializace pole s aproximací funkce sinus
- *void initADCtomA()* - inicializace pole s přepočtem hodnoty ADC na mA
- *void initOutput()* - inicializace výstupních portů
- *void initExternal50HzCounter()* - inicializace čítače T0 pro příjem externího časování
- *void stopExternal50HzCounter()* - zastavení externího časování čítače T0
- *void initADC()* - inicializace ADC převodníku
- *void initSerial()* - počáteční nastavení sériového portu
- *int16_t getOutputRMS()* - získání hodnoty výstupního napětí
- *uint16_t getOutputFreq()* - získání momentální výstupní frekvence
- *uint8_t GetReadableAmper()* - Vrací hodnotu výstupního proudu v mA
- *uint8_t GetBatteryMaxVoltage()* - Vypočte maximální hodnotu baterie z 255 naměřených hodnot

- *uint8_t GetBatteryAvgVoltage()* - Vypočte průměrnou hodnotu baterie z 255 naměřených hodnot
- *uint16_t GetOutputVoltage()* - Vrací naměřenou hodnotu výstupního napětí
- *void blockOutput()* - Odpojení baterie pomocí relé
- *void unblockOutput()* - Připojení baterie pomocí relé
- *void swapOutputs()* - Záměna výstupních portů OC1A a OC1B čítače T1
- *void switchToSteppedSinus()* - Přejchod mezi modulacemi
- *void switchToPWMSinus()* - Přejchod mezi modulacemi
- *uint8_t changeSteppedFreq(uint16_t freq)* - Změna frekvence při 50Hz PWM
- *void SendMessage(const char * message)* - Vyslání zprávy pomocí sériového portu
- *void DecodeMessage()* - Dekódování zprávy přijaté sériovým portem